



**UNIVERSITY OF CALICUT**

**Abstract**

B.Sc Programme in Computer Science – under CUCBCSS UG 2014 – Scheme and Syllabus approved - implemented w.e.f 2014 admission onwards- Orders issued.

---

**G & A - IV - J**

U.O.No. 6931/2014/Admn

Dated, Calicut University.P.O, 17.07.2014

---

*Read:-*1. U.O. No. 3797/2013/CU, dated 07.09.2013 (CBCSS UG Modified Regulations)  
(File.ref.no. 13752/GA IV J SO/2013/CU).

2. U.O. No. 5180/2014/Admn, dated 29.05.2014 (CBCSS UG Revised Regulations)  
(File.ref.no. 13752/GA IV J SO/2013/CU).

3. Item No: 1 of the minutes of BOS in Computer Science UG held on 28.1.14

4. Minutes of the meeting of Faculty of Science held on 3.2.14

5. Item II (i). of the minutes of Academic Council held on 20.3.14

**ORDER**

The Modified Regulations of Choice Based Credit Semester System for UG Curriculum w.e.f 2014 was implemented under the University of Calicut vide paper read as (1). The Revised CUCBCSS UG Regulations has been implemented w.e.f 2014 admission, for all UG programme under CUCBCSS in the University, vide paper read as (2).

As per The paper read as (3) the Scheme and Syllabus of B.Sc Programme in Computer Science under Choice based Credit Semester System (UG) was approved by the Board of Studies in Computer Science UG. As per paper read as(4) above, the Faculty of Science has approved this minutes and as per paper read as 5 above the AC has approved it.

Sanction has therefore been accorded for implementing the Scheme and Syllabus of B.Sc Programme in Computer Science with effect from 2014 admission onwards.

Orders are issued accordingly. Revised Syllabus appended.

Muhammed S  
Deputy Registrar

To

1. All Affiliated Colleges/SDE/Dept.s/Institutions under University of Calicut.
2. The Controller of Examinations, University of Calicut.

3. The Director SDE, University of Calicut.

Forwarded / By Order

Section Officer

**UNIVERSITY OF CALICUT**  
THENHIPALAM, CALICUT UNIVERSITY P.O



**DEGREE OF  
BACHELOR OF SCIENCE (B.Sc)  
IN  
COMPUTER SCIENCE  
(CHOICE BASED CREDIT AND SEMESTER SYSTEM)  
UNDER THE  
FACULTY OF SCIENCE**

**SYLLABUS**

**(FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2014 – 15 ONWARDS)**

**BOARD OF STUDIES IN COMPUTER SCIENCE (UG)  
THENHIPALAM, CALICUT UNIVERSITY P.O  
KERALA, 673 635, INDIA  
JULY, 2014**

© COPYRIGHT BY UNIVERSITY OF CALICUT, 2014

**REGULATIONS**  
**FOR DEGREE OF**  
**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**  
**(CHOICE BASED CREDIT AND SEMESTER SYSTEM)**

**EFFECTIVE FROM THE ACADEMIC YEAR 201415**

**B.SC COMPUTER SCIENCE PROGRAMME OBJECTIVE**

The basic objective of the programme is to open a channel of admission for computing courses for students, who have done the 10+2 and are interested in taking computing/IT as a career. After acquiring the Bachelor's Degree (B.Sc Computer Science) at University of Calicut, there is further educational opportunity to go for an MCA or other Master's Programme like MSc (Computer Science), MSc (IT), MBA, etc., at this university or at any other University/Institute. Also after completing the B.Sc Computer Science Programme, a student should be able to get entry level job in the field of Information Technology or ITES or they can take up self-employment in Indian & global software market. The specific objectives of the programme include

1. To attract young minds to the potentially rich & employable field of computer applications
2. To be a foundation graduate programme which will act as a feeder course for higher studies in the area of Computer Science/Applications
3. To develop skills in software development so as to enable the B.Sc Computer Science graduates to take up self-employment in Indian & global software market.
4. To train & equip the students to meet the requirements of the Software industry in the country and outside.

**PROGRAMME STRUCTURE**

**Duration:** The duration of the B.Sc Computer Science programme shall be 6 semesters distributed over a period of 3 academic years. The odd semesters (1, 3, 5) shall be from June to October and the even Semesters (2, 4, 6) shall be from November to March. Each semester shall have 90 working days inclusive of all examinations.

**Courses:** The B.Sc Computer Science programme includes four types of courses, viz., Common Courses (Code A), Core courses (Code B), Complementary courses (Code C) and Open course (Code D). The minimum number of courses required for completion of the B.Sc Computer Science programme is 37.

**Credits:** Each course shall have certain credits. For passing the B.Sc Computer Science programme the student shall be required to achieve a minimum of 120 credits of which 38 (14 for common English courses + 8 for common languages other than English + 16 credits for General courses) credit shall be from common courses, a minimum of 2 credits for project and 2 credits for the open course. Minimum credits required for core, complementary and open courses put together are 82.

**Attendance:** A student shall be permitted to appear for the semester examination, only if he/she secures not less than 75% attendance in each semester. Attendance shall be maintained by the concerned Department. Condonation of shortage of attendance to a maximum of 9 days in a semester subject to a maximum of two times during the whole period of the B.Sc Computer Science Programme may be granted by the University. Benefits of attendance may be granted to students who attend the approved activities of college/university with prior concurrence of the Head of the institution. Participation in such activities may be treated as presence in lieu of their absence on production of participation/attendance certificate in curricular/ extracurricular activities. It should be limited 9 days in a semester. The condonation of shortage of attendance shall be granted according to the existing prescribed norms.

If a student registered in first semester of the B.Sc Computer Science programme is continuously absent from the classes for more than 14 working days at the beginning of the semester without informing the authorities the matter shall immediately be brought to the notice of the Registrar of the university. The names of such students shall be removed from the rolls.

Admission to repeat courses should be within the sanctioned strength. However if more candidates are there, the candidates who have suffered serious health problems, on production of a medical certificate issued by a physician not below the rank of a Civil Surgeon in Government service, may be permitted to repeat the course, with a written order issued by the Registrar, Calicut University (by considering his/her SGPA/CGPA and percentage of attendance). The number of such candidates should not exceed two.

**Grace Marks:** Grace Marks may be awarded to a student for meritorious

achievements in co-curricular activities (in Sports/Arts/ NSS/NCC/ Student Entrepreneurship) carried out besides the regular class hours. Such a benefit is applicable and limited to a maximum of 8 courses in an academic year spreading over two semesters. No credit shall be assigned for such activities.

**Project:** Every student of the B.Sc Computer Science programme shall have to work on a project of not less than 2 credits under the supervision of a faculty member as per the curriculum.

**Extension Activities:** Compulsory social service (CSS) for a period of 15 days is essential for the successful completion of the B.Sc Computer Science programme.

## **ADMISSION**

The admission to all programmes will be as per the rules and regulations of the University. The eligibility criteria for admission shall be as announced by the University from time to time.

Separate rank lists shall be drawn up for reserved seats as per the existing rules.

The admitted candidates shall subsequently undergo the prescribed courses of study in a college affiliated to the university for six semesters within a period of not less than three years; clear all the examinations prescribed and fulfil all such conditions as prescribed by the university from time to time.

The College shall make available to all students admitted a Prospectus listing all the courses offered in various Departments during a particular semester. The information so provided shall contain title of the courses, the semester in which it is offered and credits for the courses. Detailed syllabi shall be made available in the University/college websites.

There shall be a uniform calendar prepared by the University for the Registration, conduct /schedule of the courses, examinations and publication of results. The University shall ensure that the calendar is strictly followed.

There shall be provision for inter collegiate and inter university transfer in third and fifth semester within a period of two weeks from the date of commencement of the semester. For the interuniversity or intrauniversity transfer of a student, he/she has a minimum of 20 credits in the credit bank a) in the same discipline and b) within Kerala.

Complementary changes at the time of college transfer are permitted in the third

semester if all conditions are fulfilled. Complementary changes will not be permitted in the fifth semester.

## **REGISTRATION**

Each student shall register for the courses he/she proposes to take through 'on line', in consultation with the Faculty Adviser within two weeks from the commencement of each semester. The college shall send a list of students registered for each programme in each Semester giving the details of courses registered, including repeat courses, to the university in the prescribed form within 45 days from the commencement of the semester.

A student shall be permitted to register for the examination also. If registration for examination is not possible owing to shortage of attendance beyond condonation limit, the student shall be permitted to move to the next semester. In such cases, a request from the student may be forwarded through the principal of the college to the University within two weeks of the commencement of that semester. An undertaking from the Principal may also be obtained stating that the students will be permitted to make up the shortage of attendance in that semester after completing 6 semesters. (Students shall make up the shortage of attendance in 'Repeat Semester' after completion of the programme).

The 'Repeat Semester' shall be possible only once for the entire programme and shall be done in the same college.

A student who registered for the course shall successfully complete the programme within 6 years from the year of first registration. If not, such candidate has to cancel the existing registration and join afresh as a new candidate.

The students who have attendance within the limit prescribed, but could not register for the examination have to apply for the token registration, within two weeks of the commencement of the next semester.

## **COURSE EVALUATION**

Total marks for each core, elective and open course, including lab courses and project evaluation cum programme viva voce, shall be 100 marks.

The evaluation scheme for each course shall contain two parts (1) Internal

evaluation (2) external evaluation

20% weight shall be given to the internal evaluation. The remaining 80% weight shall be for the external evaluation.

### **INTERNAL EVALUATION**

20% of the total marks in each course (i.e., 20 marks), including lab and project evaluation cum programme viva voce, are for internal examinations.

The internal assessment shall be based on a predetermined transparent system involving written test, assignments, seminars and attendance in respect of theory courses and on test/record/viva/attendance in respect of lab courses.

Components with percentage of marks of Internal Evaluation of Theory Courses are

20% of the total marks in each course (i.e., 20 marks), including lab and project evaluation cum programme viva voce, are for internal examinations.

The internal assessment shall be based on a predetermined transparent system involving written test, assignments, seminars and attendance in respect of theory courses and on test/record/viva/attendance in respect of lab courses.

Components with percentage of marks of Internal Evaluation of Theory Courses are

Test paper (50%)	– 10 Marks
Attendance (25%)	– 5 Marks
Assignment/Seminar/Viva (25%)	– 5 Marks

Components with percentage of marks of Internal Evaluation of Lab Courses are

Test paper (50%)	– 10 Marks
Attendance (25%)	– 5 Marks
Assignment/Lab involvement (25%)	– 5 Marks

Attendance of each course will be evaluated as below

Above 90% attendance	– 5 Marks
----------------------	-----------

85 to 89%	– 4 Marks
80 to 84%	– 3 Marks
76 to 79 %	– 2 Marks
75%	– 1 Marks

Internal evaluation for the project shall be generally based on content, method of presentation, final conclusion, and orientation to research aptitude. The split up shall be

Punctuality	– 4 Marks
Use of Data	– 4 Marks
Scheme/Organization of Report	– 6 Marks
VivaVoce	– 6 Marks

(If a fraction appears in internal marks, nearest whole number is to be taken)

To ensure transparency of the evaluation process, the internal assessment marks awarded to the students in each course in a semester shall be notified on the notice board at least one week before the commencement of external examination. There shall not be any chance for improvement for internal marks. The course teacher(s) shall maintain the academic record of each student registered for the course, which shall be forwarded to the University by the college Principal after obtaining the signature of both course teacher and HOD.

The marks secured for internal examination only need be sent to university, by the colleges concerned.

### **EXTERNAL EVALUATION**

There shall be University examinations for each course at the end of each semester.

Practical examinations shall be conducted by the University at the end of second, fourth and sixth semesters.

External project evaluation cum programme vivavoce shall be conducted along with the project evaluation at the end of the sixth semester.

External evaluation carries 80% of marks, i.e., 80 Marks, for each course.

External evaluation of even (2, 4 and 6) semesters will be conducted in centralized valuation camps immediately after the examination. Answer scripts of odd semester (1, 3 and 5) examination will be evaluated by home valuation. All

question papers shall be set by the university.

The model of the question paper for external examination (theory courses) of 3 Hrs duration shall be

1. **Section A** 10 compulsory objective type questions (MCQ/fill in the blank/matching/one word/etc) of 1 mark each (**Total 10 Marks**)
2. **Section B** 5 compulsory short answer type questions of 2 Marks each (either a single question or can have subdivisions) (**Total 10 Marks**)
3. **Section C** 5 short essay type questions of 4 Marks each, to be attempted from a set of 8 questions – at least one question from each unit (either a single question or can have subdivisions) (**Total 20 Marks**)
4. **Section D** 5 long essay type questions of 8 Marks each, to be attempted from a set of 8 questions – at least one question from each unit (either a single question or can have subdivisions) (**Total 40 Marks**)

The external examination in theory courses is to be conducted with question papers set by external experts. The evaluation of the answer scripts shall be done by examiners based on a well defined scheme of valuation and answer keys shall be provided by the University.

The external examination in practical courses shall be conducted by two examiners, one internal and an external, appointed by the University.

The project evaluation with programme viva voce will be conducted by two examiners, one internal and an external (appointed by the University), at the end of the sixth semester.

No practical examination will be conducted in odd semester. Practical examinations for B.Sc Computer Science programme shall be conducted in the even semester 2, 4 and 6.

The model of the question paper for external examination (lab courses) of 3 Hrs duration shall be

1. **Section A** One marked question of 30 Marks from Programming Lab Part A is to be attempted (Design Algorithm/Flowchart/Interface 10 Marks, Code 10 Marks and Result 10 Marks. **Total 30 Marks**)

2. **Section B** One marked question of 30 Marks from Programming Lab Part B is to be attempted (Design Algorithm/Flowchart/Interface 10 Marks, Code 10 Marks and Result 10 Marks. **Total 30 Marks**)
3. **Section C** Lab viva voce (**Total 10 Marks**)
4. **Section D** Lab Record (**Total 10 Marks**)

The scheme of evaluation for project cum programme viva voce shall be

1. Relevance of the Topic, Statement of Objectives, Methodology (Reference/Bibliography) (**Total 16 Marks**)
2. Presentation, Quality of Analysis/Use of Statistical tools, Findings and recommendations (**Total 24 Marks**)
3. Project cum Programme Viva Voce (**Total 40 Marks**)

## **REVALUATION**

In the new system of grading, revaluation is permissible. The prevailing rules for revaluation are applicable.

Students can apply for photocopies of answer scripts of external examinations. Applications for photocopies/scrutiny/revaluation should be submitted within 10 days of publication of results. The fee for this shall be as decided by the university.

## **COURSE IMPROVEMENT**

A maximum of two courses (Common, Core, Complementary or Open) can be improved in each semester. Improvement of a particular semester can be done only once. The student shall avail the improvement chance in the succeeding year after the successful completion of the semester concerned. The internal marks already obtained will be carried forward to determine the grades/marks in the improvement examination. If the candidate fails to appear for the improvement examination after registration, or if there is no change in the results of the improvement examination appeared, the marks/grades obtained in the first appearance will be retained.

Improvement and supplementary examinations cannot be done simultaneously.

## EVALUATION AND GRADING

Mark system is followed instead of direct grading for each question (for both internal and external examinations). For each course in the semester letter grade, grade point and % of marks are introduced in 7 point indirect grading system. The grading on the basis of a total internal and external mark will be indicated for each course and for each semester and for the entire programme.

Indirect Grading System in 7 point scale is as below

% of Marks	Grade	Interpretation	Grade Point (G)	Range of Grade Points	Class
90 and above	A+	Outstanding	6	5.5 to 6	First class with distinction
80 to below 90	A	Excellent	5	4.5 to 5.49	
70 to below 80	B	Very Good	4	3.5 to 4.49	First class
60 to below 70	C	Good	3	2.5 to 3.49	
50 to below 60	D	Satisfactory	2	1.5 to 2.49	Second class
40 to below 50	E	Pass/Adequate	1	0.5 to 1.49	Pass
Below 40	F	Failure	0	0 to 0.49	Fail

An aggregate of E grade with 40% marks (after external and internal put together) is required in each course for a pass and also for awarding a degree.

Appearance for Internal Assessment (IA) and End Semester Evaluation (ESE external) are compulsory and no grade shall be awarded to a candidate if she/he is absent for IA/ESE or both. For a pass in each course 40% marks or E grade is necessary

A student who fails to secure a minimum grade for a pass in a course is permitted to write the examination along with the next batch.

After the successful completion of a semester, Semester Grade Point Average (SGPA) of a student in that semester is calculated using the formula given below. For the successful completion of a semester, a student should pass all courses. However, a student is permitted to move to the next semester irrespective of SGPA obtained.

The Semester Grade Point Average can be calculated as

$$SGPA = \frac{\text{Sum of the credit points of all courses in a semester}}{\text{Total Credits in that semester}}$$

$$i. e., \quad SGPA = \frac{C1 \times G1 + C2 \times G2 + C3 \times G3 + \dots}{n}$$

where  $G1, G2, \dots$  are grade points of different courses;  $C1, C2, \dots$  are credits of different courses of the same semester and  $n$  is the total credits in that semester.

The Cumulative Grade Point Average (CGPA) of the student is calculated at the end of a programme. The CGPA of a student determines the overall academic level of the student in a programme and is the criterion for ranking the students. CGPA can be calculated by the following formula

The Cumulative Grade Point Average (CGPA) can be calculated as

$$CGPA = \frac{\text{Total credit points obtained in six semesters}}{\text{Total Credits}}$$

## **GRADE CARD**

The University shall issue to the students grade/marks card (by online) on completion of each semester, which shall contain the following information

- a. Name of University
- b. Name of college
- c. Title of Undergraduate Programme
- d. Semester concerned
- e. Name and Register Number of student
- f. Code number, Title and Credits of each course opted in the semester
- g. Internal marks, External marks, total marks, Grade point (G) and letter grade for each course in the semester
- h. The total credits, total credit points and SGPA in the semester (corrected to two decimal places)
- i. Percentage of total marks

The final Grade/mark Card issued at the end of the final semester shall contain the details of all courses taken during the entire programme including those taken over and above the prescribed minimum credits for obtaining the degree. However, for the compilation of CGPA only the best performed courses, if any, with maximum grade points alone shall be taken subject to the minimum credits requirements

(120) for passing a specific degree. The final grade card shall show the percentage of marks, CGPA (corrected to two decimal places) and the overall letter grade of a student for the entire programme. The final grade/mark card shall also include the grade points and letter grade of common courses, core courses, complementary courses and open courses, separately. This is to be done in a seven point indirect scale.

## **AWARD OF DEGREE**

The successful completion of all the courses (common, core, complementary and open courses) prescribed for the B.Sc Computer Science programme with E grade (40 %) shall be the minimum requirement for the award of B.Sc Computer Science programme degree.

## **GRIEVANCE REDRESSAL COMMITTEE**

### **COLLEGELEVEL**

The College shall form a Grievance Redressal Committee in each department comprising of course teacher and one senior teacher as members and the Head of the department as Chairman. This committee shall address all grievances relating to the internal assessment grades of the students. There shall be a CollegeLevel Grievance Redressal Committee comprising of Student Advisor, two senior teachers and two staff council members (one shall be elected member) as members and principal as Chairman.

### **UNIVERSITYLEVEL**

The University shall form a Grievance Redressal Committee as per the existing norms.

## **B.SC. COMPUTER SCIENCE PROGRAMME STRUCTURE**

### **SEMESTER I**

Course No.	Course Code	Title	Hours		Marks		Credit
			T	P	Int.	Ext.	
01	XXXXA01	Communication Skills in English	5	0	20	80	4
02	XXXXA02	Critical Reasoning, Writing and Presentation	4	0	20	80	3
03	XXXXA07	Communication Skill in Languages other than English	4	0	20	80	4
04	BCS1B01	Problem Solving Using C	2	2	20	80	3
05	XXXXC01	Complementary Mathematics I	4	0	20	80	3
06	XXXXC01	Optional Complementary I*	4	0	20	80	3
<b>Total</b>			<b>25</b>		<b>600</b>		<b>20</b>

*\*Hours distribution, Mark distribution and credits may be different for the optional complementary course with laboratory works*

### **SEMESTER II**

Course No.	Course Code	Title	Hours		Marks		Credit
			T	P	Int.	Ext.	
07	XXXXA03	Reading Literature in English	5	0	20	80	4
08	XXXXA04	Reading on Indian Constitution Secularism and Sustainable Environment	4	0	20	80	3
09	XXXXA08	Literature in Languages other than English	4	0	20	80	4
10	BCS2B02	OOP Concepts and Data Structures Using C++	2	0	20	80	3
11	BCS2B03	Programming Laboratory I: Programming in C & Data	0	2	20	80	2

		Structures Using C++					
12	XXXXC02	Complementary Mathematics II	4	0	20	80	3
13	XXXXC02	Optional Complementary II*	4	0	20	80	3
<b>Total</b>			<b>25</b>		<b>700</b>		<b>22</b>

*\*Hours distribution, Mark distribution and number of courses and their credits may be different for the optional complementary course with laboratory works*

### SEMESTER III

Course No.	Course Code	Title	Hours		Marks		Credit
			T	P	Int.	Ext.	
14	XXXXA11	Numerical Skill	4	0	20	80	4
15	XXXXA12	General Informatics	4	0	20	80	4
16	BCS3B04	Fundamentals of Digital Electronics	2	1	20	80	3
17	BCS3B05	Visual Programming Using VB.NET	1	3	20	80	3
18	XXXXC02	Complementary Mathematics III	5	0	20	80	3
19	XXXXC02	Optional Complementary III*	5	0	20	80	3
<b>Total</b>			<b>25</b>		<b>600</b>		<b>20</b>

*\*Hours distribution, Mark distribution and number of courses and their credits may be different for the optional complementary course with laboratory works*

### SEMESTER IV

Course No.	Course Code	Title	Hours		Marks		Credit
			T	P	Int.	Ext.	
20	XXXXA13	Entrepreneurship	4	0	20	80	4
21	XXXXA14	Basics of Audio & Video Media	4	0	20	80	4
22	BCS4B06	Fundamentals of Database Management System and RDBMS	3	2	20	80	3
23	BCS4B07	Programming Laboratory II: VB.NET & RDBMS	0	2	20	80	2
24	XXXXC03	Complementary Mathematics IV	5	0	20	80	3
25	XXXXC03	Optional Complementary IV*	5	0	20	80	3
<b>Total</b>			<b>25</b>		<b>600</b>		<b>19</b>

*\*Hours distribution, Mark distribution and number of courses and their credits may be different for the optional complementary course with laboratory works*

### SEMESTER V

Course No.	Course Code	Title	Hours		Marks		Credit
			T	P	Int.	Ext.	
26	BCS5B08	Computer Organization and Architecture	5	0	20	80	4
27	BCS5B09	Java Programming	3	3	20	80	4
28	BCS5B10	Web Programming Using PHP	3	3	20	80	4
29	BCS5B11	Principles of Software Engineering	4	0	20	80	4
30	XXX5DXX	Open Course	2	0	10	40	2
		Project Work	0	2			
<b>Total</b>			<b>25</b>		<b>450</b>		<b>18</b>

## SEMESTER VI

Course No.	Course Code	Title	Hours		Marks		Credit
			T	P	Int.	Ext.	
31	BCS6B12	Android Programming	4	1	20	80	4
32	BCS6B13	Fundamentals of Operating Systems	4	1	20	80	4
33	BCS6B14	Computer Networks	5	0	20	80	4
34	BCS6B15	Programming Laboratory III: Java & PHP Programming	0	2	20	80	2
35	BCS6B16	Programming Laboratory IV: Android & Linux shell Programming	0	2	20	80	2
36	BCS6B17X	Elective Course	4	0	20	80	3
37	BCS6B18	Project Work	0	2	10	40	2
<b>Total</b>			<b>25</b>		<b>650</b>		<b>21</b>

## LIST OF OPEN COURSES

Course No.	Course Code	Title
31	BCS5D01	Introduction to Computers & Office Automation
31	BCS5D02	Web Designing
31	BCS5D03	Introduction to Problem Solving and C Programming

## LIST OF ELECTIVE COURSES

Course No.	Course Code	Title
37	BCS6B17a	Computer Graphics
37	BCS6B17b	System Software
37	BCS6B17c	Microprocessor & Applications

## MARK DISTRIBUTION

1	Common: English (4 Courses×100 Marks)	400
2	Additional: Mal/Hindi... (2 Courses×100 Marks)	200
3	General (4 Courses×100 Marks)	400
4	Core (Theory & Practical) (17 Courses×100 Marks)	1700
5	Project& Viva-Voce	50
6	Open (1 Course)	50
7	Complementary (8 Courses×100 Marks*) <i>(*Number of complementary courses and their mark distribution may vary for complementary courses with laboratory work)</i>	800
<b>Total Marks</b>		<b>3600</b>
Total Courses <i>(37 Courses for Optional Complementary without Lab 38 Courses for Optional Complementary with Lab)</i>		37/38



## FIRST SEMESTER

### BCS1B01: Problem Solving Using C

**Course Number:** 4

**Contact Hours per Week:** 4 (2T + 2P)

**Number of Credits:** 3

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Aim of the Course

- To equip the students with fundamental principles of Problem Solving aspects.
- To learn the concept of programming
- To study C language
- To equip the students to write programs for solving simple computing problems

#### Prerequisites

- Background of the basic science at +2 level

#### Course Outline

##### Module I [6T+6P]

Introduction: The problem solving aspect, Top-down design, Implementation of algorithms, Program verification, efficiency of algorithms. Introduction to C Programming, overview and importance of C, C Program Structure and Simple programs, Creation and Compilation of C Programs under Linux and Windows Platforms.

##### Module II [6T+6P]

Elements of C Language and Program constructs: Character Set, C Tokens, Keywords and Identifier, Constants, Variables, Data types, Variable declaration and assignment of values, Symbolic constant definition. C Operators, Arithmetic operators, relational operators, and logical operators, assignment operators, increment and decrement operators, conditional operators, special operators, arithmetic expressions, evaluation of expressions, precedence of arithmetic

operators, Type conversion in expressions, operator precedence and associativity, Mathematical Functions, I/O operations.

### **Module III [6T+6P]**

Decision making, Branching and Looping. Decision making with IF statement, Simple IF statement, Ifelse statement, Nesting of Ifelse and elseif Ladder, Switch statement, Conditional operator, Goto statement. Looping While loop, Do-While, and For Loops, Nesting of loops, jumps in loop, skipping of loops.

### **Module IV [6T+6P]**

Array & Strings One dimensional array, two dimensional array and multidimensional array, strings and string manipulation functions. The Concept of modularization and User defined functions, Multifunction Program, calling functions, various categories of functions, Nesting of functions and recursion, functions and arrays, scope and lifetime of variables in functions, multi-file programs. Structures & Union structure definition, giving values to members, structure initialization, comparison of structure variables, arrays of structures, arrays within structures, structures within arrays, structures and functions, Unions, bit fields.

### **Module V [6T+6P]**

Pointers and Files Understanding pointers, accessing the address of a variable, declaring and initializing pointers, accessing a variable through its pointer, pointer expressions, pointer and arrays, pointer and character string, pointers and functions, pointers and structures, pointer to pointer dynamic memory allocation.

Files Defining, Opening and closing files I/O operations on files error handling on files random access of files command line operations. Preprocessor directives, Macro substitution directives, simple macros, macros with arguments, nesting of macros, Compiler control directives.

### **References**

1. E. Balaguruswami, *Programming in ANSI C*
2. Brian W. Kernighan & Dennis M. Ritchie, *The C Programming Language*
3. Yashavant P. Kenetkar, *Let us C*
4. Byran Gotfried, Schaums Outline series *Programming with C*
5. Ashok N. Kamthane, *Programming in C*, Pearson, 2nd Edition.

## SECOND SEMESTER

### BCS2B02: OOP Concepts and Data Structures Using C++

**Course Number:** 10

**Contact Hours per Week:** 4 (2T + 2P)

**Number of Credits:** 2

**Number of Contact Hours:** 30 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Aim of the Course

- To learn basic concepts of OOPS
- To learn Object Oriented Programming through C++
- To introduce the concept of data structures
- To make the students aware of various data structures
- To equip the students implement fundamental data structures

#### Prerequisites

- Knowledge in C Programming Language

#### Course Outline

##### Module I [6T]

C++ Fundamentals: Introduction: Characteristics and principles of OOP, C++ Fundamentals: C++ data types, Operators, Expressions, Type conversion, iostream library, Control statements, Functions: Prototype, Arguments passing, Return type, Default arguments, Inline functions, Function overloading Classes: Classes and Objects, Defining classes, Creating objects, Defining member function, Static class members, Friend functions, Passing and returning objects to and from functions, Nesting of classes Constructors: Default constructors, Parameterized constructors, Constructor overloading, Constructors with default arguments, Copy constructors - Destructors.

## **Module II [6T]**

Advanced C++ Features: Pointers, Dynamic memory management, new and delete operators, Pointers to objects, Pointers to object members, Accessing members, this pointer, Operator overloading: Overloading unary and binary operators, Type conversion: Between objects and basic types and between objects of different classes, Inheritance: Single Inheritance, Overriding base class members, Abstract classes, Constructors and destructors in derived classes, Multilevel inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Virtual functions, Virtual base class, File processing: Opening and closing files, File pointers, Filestream functions, Creating and processing text and binary files.

## **Module III [6T]**

Contiguous Data Structures: Arrays: Structure of arrays, Representation of arrays, Operations on one dimensional arrays, Overloading operators for one-dimensional arrays, Multidimensional arrays, String representation and manipulation.

## **Module IV [6T]**

Non-Contiguous Linear Data Structures: Non-contiguous Data Structures: Lists: Representation and Traversing of linked list, Operations with linked list, Doubly linked list, Circular list, Stacks: Definition, Operation on stack, Implementation using arrays and linked lists, Evaluation of arithmetic expressions, Queues: Definition, Implementations using arrays and linked lists, Circular queue, Applications of queues.

## **Module V [6T]**

Searching and Sorting: Searching: Linear search, Binary search, Comparison of different methods, Sorting: Insertion, Bubble, Selection, Quick and Merge sort methods, Comparisons, Hashing: Different hashing functions, Methods for collision handling.

## **References:**

1. E. Balaguruswamy, *Object oriented programming with C++*

2. Robert C Lafore, *Object oriented programming* with C++
3. Bjarne Stroustrup, *The C++ Programming Language*, Addison Wesley.
4. E. Horowitz & S. Sahni, *Fundamentals of data structures*
5. Aron M, Tenenbaum, *Data Structure Using C and C++*

## **CSC2B03: Programming Laboratory I: Programming in C &Data Structure Using C++**

**Course Number:** 11

**Contact Hours per Week:** 2P

**Number of Credits:** 3

**Number of Contact Hours:** 30 P.Hrs.

**Course Evaluation:** Internal – 20 Marks + External –80 Marks

### **Aim of the Course**

- To make the students learn programming environments.
- To practice procedural/OO programming concepts.
- To make the students equipped to solve mathematical or scientific problems using C/C++.
- To learn how to implement various data structures.
- To provide opportunity to students to use data structures to solve real life problems.

### **Prerequisites**

- Knowledge in operating computer.
- Theoretical knowledge in Data structures.
- Knowledge in C/C++ Programming.

### **Course Outline:**

#### **Part A: C Programming**

Experiments should include but not limited to:

- Lab 1: Simple C Programs like Computation of area of a circle, rectangle etc.
- Lab 2: C Programs using Loops (like pyramid printing, factorial computation, number reversing etc.)
- Lab 3: Programs involving Arrays (Searching, sorting and vector operations etc.)
- Lab 4: Two dimensional arrays (like Matrix operations)
- Lab 5: String Manipulations
- Lab 6: Programs involving Structures (like addition of Two Complex numbers, student record creation and manipulation etc.)
- Lab 7: Programs involving Union
- Lab 8: Programs involving functions
- Lab 9: Recursion (recursive function to compute a factorial, reverse string etc)
- Lab 10: Pointers - simple programs to learn concept of pointers, array operation using pointers etc.
- Lab 11: File operations
- Lab 12: Command line arguments copy a file, delete a file etc.

**Projects:**

1. Design a Scientific Calculator and include as many functions as possible
2. Prepare rank list for finding the topper in the class considering the grade obtained in various subjects.
3. Develop a telephone directory.
4. Write a program for sorting a list of input elements. User should be able to give an option for ascending order or descending order. Use command line arguments.

*Note: All lab works should be neatly recorded in a Laboratory Record Book in written form. However Program results can be pasted in the left hand side of the fare record. All students should have a rough record (observation note book) too, in which they write all the works to be carried out in the lab prior to his/her entering the lab. He/She may also note down the i/p and o/p that he gives for program verification in the observation note book (rough record).*

## **Part B: Data Structure Using C++**

Experiments should include but not limited to:

- Lab 1: Implementation of Object-Oriented concepts like classes, objects, overloading, constructors, destructors, inheritance, etc
- Lab 2: Implementation of array operations
- Lab 3: Stacks operation using array
- Lab 4: Stacks: adding, deleting elements using linked list
- Lab 5: Circular Queue: Adding & deleting
- Lab 6: Implementation of linked lists: inserting, deleting, inverting a linked list.
- Lab 7: Implementation of queues using linked lists
- Lab 8: Implementation Polynomial addition, Polynomial multiplication using linked lists.
- Lab 9: implementation of doubly linked list
- Lab 10: implementation of queue using array
- Lab 11: Implementation of searching Techniques: Linear and Binary search.
- Lab 12: Sorting techniques: Bubble Sort, Insertion Sort, Quick Sort, and Merge Sort.
- Lab 13: Simple Hashing Functions.

*Note: All lab works should be neatly recorded in a Laboratory Record Book in written form. However Program results can be pasted in the left hand side of the fare record. All students should have a rough record (observation note book) too, in which they write all the works to be carried out in the lab prior to his/her entering the lab. He/She may also note down the i/p and o/p that he gives for program verification in the observation note book (rough record).*

## THIRD SEMESTER

### BCS3B04: Fundamentals of Digital Electronics

**Course Number:** 16

**Contact Hours per Week:** 3 (2T + 1P)

**Number of Credits:** 3

**Number of Contact Hours:** 48 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Objectives of the Course

- To learn number systems and boolean algebra
- To learn combinational and sequential circuits
- To learn A/D and D/A converters

#### Prerequisites

Basic Mathematical skill

#### Course Outline

##### Module I [7T+3P]

Number Systems and Codes, Decimal numbers, binary numbers, binary arithmetic, 1's and 2's complements, octal numbers, hexadecimal numbers, interconversions of number systems, Digital codes: Binary coded decimal(BCD), Gray code, Excess-3 code, ASCII code, error detection and error correction codes, Hamming code.

##### Module II [7T+3P]

Logic Gates Positive and negative logic, NOT gate, OR gate, AND gate, NAND gate, NOR gate, EX-OR and EX-NOR gates, Universal gates. Boolean Algebra: Boolean operations, logic expressions, rules and laws of Boolean algebra, DeMorgan's theorems, minterms, maxterms, SOP and POS form of Boolean expressions for gate network, simplification of Boolean expressions using Boolean algebra and Karnaugh map techniques (up to 4 variables)

**Module III [6T+3P]**

Arithmetic and Combinational Logic Circuits Half adder, full adder, parallel binary adder, decoders, BCD to 7-segment decoder, multiplexers and demultiplexers, multiplexer and demultiplexer trees.

**Module IV [6T+3P]**

Sequential Logic Circuits: SR latch, SR flip flop, JK flip flop, Master Slave JK flip flop, D type flip flop, T type flip flop. Shift register: serial in - serial out, serial in - parallel out, parallel in - serial out, parallel in-parallel out configurations. Ring counter, Johnson's counter, asynchronous counters, synchronous counters, up/down asynchronous counter,

**Module V [6T+4P]**

A/D and D/A converters: D/A conversions – Weighted-Register D/A converter, R-2R ladder D/A converter, A/D conversions-Counter type method using D/A, dual slope integrator method, successive approximation method, simultaneous method.

**References:**

1. Rajaraman V. & Radhakrishnan, *An Introduction to Digital Computer Design*, PHI.
2. Thomas L Floyd, *Digital Fundamentals*, Universal Book Stall
3. Malvino & Leach, *Digital Principles & Applications*, TMH
4. Jain R.P. , *Modern Digital Electronics*, TMH
5. Malvino, *Digital Computer Electronics*, TMH
6. Bartee T.C., *Digital Computer Fundamentals*, THM
7. William H. Gothmann, *Digital Electronics: An Introduction to Theory and Practice*, PHI

**BCS3B05: Visual Programming Using VB.NET**

**Course Number:** 17

**Contact Hours per Week:** 4 (1T + 3P)

**Number of Credits:** 3

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Objectives of the Course**

- To get a general understanding on .Net Frame Work
- To get a general understanding on ADO.Net

### **Prerequisites**

Basic knowledge of OOP

### **Course Outline**

#### **Module I [3T+9P]**

Introduction to visual programming - Concept of event driven programming - Introduction to VB.Net environment, The .NET Framework and the Common Language Runtime. Building VB.NET Applications, The Visual Basic Integrated Development - Basic Language - Console application and windows application, Data types, Declaring Variables, scope of variables, operators and statements.

#### **Module II [3T+9P]**

Making Decisions with If . . . Else Statements, Using Select Case, Making Selections with Switch and Choose, Loop statements - Do Loop, for, while - The With Statement - Handling Dates and Times - Converting between Data Types - Arrays - declaration and manipulation - Strings & string functions - Sub Procedures and Functions.

#### **Module III [4T+8P]**

Windows Applications - Forms - Adding Controls to Forms, Handling Events, MsgBox , InputBox , Working with Multiple Forms, Setting the Startup Form, SDI & MDI Forms, Handling Mouse & Keyboard Events, Common controls (Text Boxes, Rich Text Boxes, Labels, Buttons, Checkboxes, Radio Buttons, Group Boxes, List Boxes, Checked List Boxes, Combo Boxes, Picture Boxes, Scroll Bars, Tool Tips, Timers) properties - methods

### **Module IV [4T+8P]**

Object-Oriented Programming - Creating and using Classes & objects - Handling Exceptions - On Error GoTo - Raising an Exception - Throwing an Exception - Using Structured Exception Handling - Debugging and tracing.

### **Module V [4T+8P]**

Data Access with ADO.NET - Accessing Data with the Server Explorer - Accessing Data with Data Adaptors and Datasets - Creating a New Data Connection - Creating and populating Dataset - Displaying Data in a Data Grid - Selecting a Data Provider - Data Access Using Data Adapter Controls - Binding Data to Controls - Handling Databases in Code - Binding to XML data.

### **References:**

1. Steven Holzner, *Visual Basic .NET Black Book*
2. Rebecca Riordan, *VB.NET for Developers*, Keith Franklin, SAMS
3. Jason Beres, *Sams Teach Yourself Visual Studio .NET 2005 in 21 Days*,
4. Jesse Liberty, *Learning Visual Basic .NET*
5. Tim Anderson, *Visual Basic .Net programming in Easy Steps*, DreamTech Press.
6. *.NET Programming (6-in-1), Black Book*, Kogent Learning Solutions Inc., Wiley- Dream Tech Press.
7. Francisco, *Visual Studio .Net*, Microsoft Publication
8. *.Net Framework Essentials*, O'Reilly

## FOURTH SEMESTER

### BCS4B06: Fundamentals of Database Management System and RDBMS

**Course Number:** 22

**Contact Hours per Week:** 5 (3T + 2P)

**Number of Credits:** 3

**Number of Contact Hours:** 75 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Aim of the Course

- To learn the basic principles of database and database design
- To learn the basics of RDBMS
- To learn the concepts of database manipulation SQL
- To study PL/SQL language

#### Prerequisites

Basic knowledge of computers, data structures and programming

#### Course Outline

##### Module I [10T+5L]

Database System concepts and applications Introduction to databases, File Systems vs DBMS, Advantages and Disadvantages of using DBMS Approach, Database administrators and user, Data Models, Schemas, and Instances, Types of Data Models, Three Schema Architecture and Data Independence, Database Languages and Interfaces. Conceptual Data Models for Database Design Entity Relationship Models, Concept of Entity, Entity Sets, Relationship Sets, Attributes, Domains, Constraints, Keys, Strong and Weak Entities. Concepts of EER

**Module II [10T+5L]**

Relational Data Model Relations, Domains and Attributes, Tuples, Keys. Integrity Rules, Relational Algebra and Operations, Relational Calculus and Domain Calculus. Relational Database Design using ER to Relational Mapping. SQL Data Definition in SQL creation, updation, deletion of tables, modifying the structure of tables, renaming, dropping of tables. Constraints. Database Manipulation in SQL Select command, Logical operators, Range searching, Pattern matching, Grouping data from tables in SQL, GROUP BY, HAVING clauses, Joins Joining Multiple Tables, Joining a Table to itself. Views Creation, Renaming the column of a view, destroys view.

**Module III [10T+5L]**

Relational database design Anomalies in a Database, Normalization Theory, Functional Dependencies. First, Second and Third Normal Forms, Relations with more than one Candidate Key, Good and Bad Decompositions, Boyce Codd Normal Form, Multivalued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

**Module IV [9T+6L]**

Transaction Management and Concurrency Control Transaction Properties (ACID), states, Commit, Rollback; Concurrency Control Lost update problems, Locks, two phase locking.

**Module V [9T+6L]**

Programming with SQL: Data types, Using set and select commands, procedural flow, if, if /else, while, goto, global variables, Security: Locks, types of locks, levels of locks. Cursors: Working with cursors, Error Handling, Developing stored procedures, create, alter and drop, passing and returning data to stored procedures, using stored procedures within queries, building user defined functions, creating and calling a scalar function, implementing triggers, creating triggers, multiple trigger interaction.

### References:

1. Abraham Silberschatz, Henry F Korth, S.Sudharshan, *Database System Concepts*
2. Ivan Bayross, *PL/SQL: The Programming Language of Oracle SQL*.
3. Alex Krigel and Boris M.Trukhnov, *SQL Bible*, Wiley pubs
4. Paul Nielsen, *Microsoft SQL Server 2000 Bible*, Wiley Dreamtech India Pubs.
5. CJ Date, *Introduction to Database Systems*, Addison Wesley.
6. Ramkrishnan, *Database Management Systems*, McGraw Hill.

### BCS4B07: Programming Laboratory II: VB.NET & RDBMS

**Course Number:** 23

**Contact Hours per Week:** 2 (0T + 2P)

**Number of Credits:** 2

**Number of Contact Hours:** 45 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### Aim of the Course

- To practice Visual Programming using VB.NET.
- To learn practical database design.
- To create & manipulate various database objects.
- To practice SQL Commands.

### Prerequisites

- Theoretical knowledge in Visual Programming.
- Theoretical knowledge in DBMS & SQL.

**Course Outline:**

**Part A: VB.NET**

- Lab 1. Program in different kinds of arrays including jagged arrays.
- Lab 2. Program using structure and enum
- Lab 3. Program using classes, methods, properties and read only property
- Lab 4. Program using constructors, overload constructors and class events
- Lab 5. Program using exception handling
- Lab 6. Program to perform various string operations
- Lab 7. Program using .net built-in collection classes namely array list, CSC array, hash table, queue, sorted list, stack, collection, dictionary base.
- Lab 8. Program using inheritance, constructors in inheritance
- Lab 9. Program using overriding, abstract base classes, shared members and interface
- Lab 10. Program using streams and serialization
- Lab 11. Programs to demonstrate Control Statements
- Lab 12. Programs to demonstrate Controls
- Lab 13. Programs to demonstrate Dialog Boxes
- Lab 14. Programs to demonstrate Mouse Events
- Lab 15. Programs to demonstrate Keyboard Events
- Lab 16. Programs to demonstrate Class Concept
- Lab 17. Programs to demonstrate Inheritance
- Lab 18. Programs to demonstrate data access using Data Binding Controls
- Lab 19. Programs to demonstrate data access with SQLServer using code

## **Part B: DBMS Lab**

**Lab 1:** In this session you need to create database for an Employee management system of an ABC organization. The details about different tables are given below. According to that you can proceed further and create tables using PostgreSQL/ MySQL/PL/SQL.

Create the following tables with the specified constraints:

### **Department:**

Department name - Not NULL unique

Department number - Primary Key

ManagerId - Refers to employee-id of employee table.

Manager

date of joining - Not NULL.

### **Employee:**

First name - Not NULL

Middle initials

Last name - Not NULL

Employee id - Primary Key

Date of Birth - Not NULL

Address

Gender - M or F

Salary - Range of 5000 to 25000

Date of Joining

Department number - Refers to Department Number of Department table.

### **Department location:**

Department number - Refers to Department number of department table.

Department location - Not NULL.

Department number & Department location are combined Primary Key

### **Project:**

Project name-Not NULL.

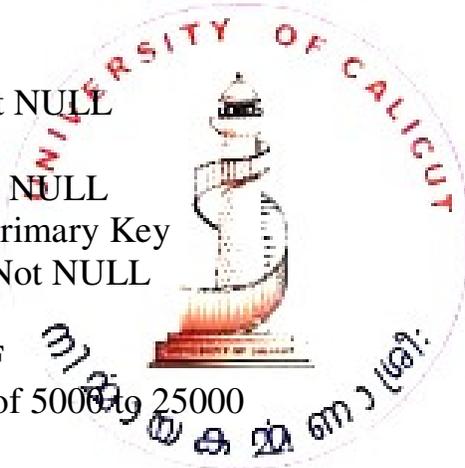
Project number-Primary Key.

Project location-Not NULL.

Department number-Refers to department number of Department table.

### **Works-on:**

Employee-id - Not NULL refers to employee-id of employee table.



Project number- Not NULL refers to Project number of Project table.

Hours - Not NULL.

Employee-id & Project number are combined primary key.

**Dependent:**

Employee-id - Refer to employee table employee id field

Dependent name - Gender - M or F

Date of Birth - Not NULL

Relationship - Not NULL

Now enter a few sets of meaningful data and answer the following queries.

1. List the department wise details of all the employees.
2. Find out all those departments that are located in more than one location.
3. Find the list of projects.
4. Find out the list of employees working on a project.
5. List the dependents of the employee whose employee id is 001

**Lab 2:** These sessions is similar to the previous one, but in this session, assume that you are developing a prototype database of the College library management system for that you need to create the following tables:

- Book Records
- Book details
- Member details and
- Book issue details

**Book Records:**

Accession Number

ISBN Number

**Books:**

ISBN Number

Author

Publisher

Price

**Members:**

Member Id

Member Name

Maximum Number of books that can be issued

Maximum Number of days for which book can be issued

**Book Issue:**

Member Id

Accession Number

Issue Date

Return Date

You must create constraints, including referential integrity constraints, as appropriate. Please note accession number is unique for each book. A book, which has no return date, can be considered as issued book. Enter suitable data into the tables. Now answer the following:

1. Insert data in all the three tables (use insert).
2. Insert appropriate description associated with each table and the column (use comment).
3. Display the structure of the tables.
4. Display the comments that you have inserted.
5. Using SELECT statement, write the queries for performing the following function:
  - (a) Get the list of all books (No need to find number of copies).
  - (b) Get the list of all members.
  - (c) Get the Accession number of the books which are available in the library.
  - (d) On return of a book by a member calculate the fine on that book.
  - (e) List of books issued on 01-Jan-2005.
  - (f) Get the list of all books having price greater than Rs. 500/-
  - (g) Get the list of members who did not have any book issued at any time.
  - (h) Get the list of members who have not returned the book.

- (i) Display member ID and the list of books that have been issued to him/her from time to time.
- (j) Find the number of copies of each book (A book accession number would be different but ISBN number would be the same).
- (k) Find the number of copies available of a book of given ISBN number.
- (l) Get the member ID and name of the members to whom no more books can be issued, because they have already got as many books issued as the number for which they are entitled.

**Lab 3:** This session is based on Lab 2 where you have created a library management system. In this session you have different query specification. You must create appropriate forms, reports, graphs, views and data filtering, use of multilevel report, etc to answer these queries.

1. Get the list of ISBN-Number, Book name, available copies of the books of which available copies are greater than zero.
2. Get the list of ISBN-Number, Book name, Total copies, available copies of the book of which available copies are greater than zero. List should be displayed in alphabetical order of book name.
3. Get the list of ISBN number, Book name, Author, total copies, cost (cost is price total copies). List should be displayed in descending order of cost.
4. Get the list of books issued to each member.
5. Write query to know the maximum and average price of the books.
6. Get the list of all existing members and the number of days for which a member is allowed to keep the book. Also find out the members who have got the maximum number of books issued.
7. Get the list of member codes of those members who have more than two books issued.
8. Find the details of the books presently issued to a member.
9. Create the history of issue of a book having a typical accession number.
10. To set the width of the book name as 35.

**Lab 4:** Create the following table and perform the necessary tasks defined below

one by one. You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ using client/server wherever needed.

1. Create the following table named customer

Column name	type	size
Customer id	Character	10
Name	Character	25
Area	Character	3
Phone	Numeric	7

Insert the appropriate data into table and do the following.

- Update Phone numbers of all customers to have a prefix as your city STD Code
- Print the entire customer table
- List the names of those customers who have e as second letter in their names.
- Find out the Customer belonging to area abc
- Delete record where area is NULL.
- Display all records in increasing order of name.
- Create a table temp from customer having customer-id, name, and area fields only
- Display area and number of records within each area (use GROUP by clause)
- Display all those records from customer table where name starts with a or area is abc.
- Display all records of those where name starts with a and phone exchange is 55.

**Lab 5:** Answer the following queries using Library system as created earlier. You must create a view to know member name and name of the book issued to them, use any inbuilt function and operators like IN, ANY, ALL, EXISTS.

- a. List the records of members who have not been issued any book using EXISTS operator.
- b. List the members who have got issued at least one book (use IN /

ANY operator).

- c. List the books which have maximum Price using ALL operator.
- d. Display Book Name, Member Name, Issue date of Book. Create a view of this query of the currently issued books.

**Lab 6:** Create a table of Employee (emp-number, name, dept, salary) and Department (dept number, dept name). Insert some records in the tables through appropriate forms having integrity checks. Add some records in employee table where department value is not present in department table. Now answer the following query:

- a. Display all records from employee table where department is not found in department table.
- b. Display records from employee table in a report format with proper headings. This report must also contain those records where department number does not match with any value of department table.
- c. Display those employee records who have salary less than the salary of person whose empcode= A100.
- d. Create another table: SalesData (RegionCode, City, Salesperson-Code, SalesQty).
- e. Display records where salesperson has achieved sales more than average sales of all sales persons of all the regions.

**Lab 7:** Create the following tables:

Order party: (Order number, Order date, customer code)

Order: Order number, Item code, Quantity

The key to the second table is order-number + item-code

Create a form for data entry to both the tables.

**Lab 8:** Implement student information system

**Lab 9:** Implement hotel billing system

**Lab 10:** SQL scripts to display various reports like Result of an Examination, Salary Report, Sales Report, Sales reports grouped on Sales person or item, etc

**Lab 11:** Write simple PL/SQL anonymous blocks for displaying whole numbers from 1 to 100, odd numbers from 1 to 100, even numbers from 1 to 100, positive whole numbers up to a given number, odd numbers from 1 to a given number, even numbers from 2 to a given number, Fibonacci

numbers up to 100, Strange numbers up to 1000, factorials of the numbers from 1 to 10, etc.

**Lab 12:** Write simple PL/SQL anonymous blocks using functions for displaying Fibonacci numbers up to a given number, strange numbers up to a given number, factorials of numbers up to a given number, etc

**Lab 13:** Create a STUDENT table with following fields:

Field Name	Type	Width
regno	character	10
name	character	20
paper1	numeric	3
paper2	numeric	3
paper3	numeric	3
paper4	numeric	3
paper5	numeric	3
total	numeric	3
result	character	6
grade	character	1

Enter the regno, name and marks in 5 papers of at least 10 students. Write a PL/SQL program to process the records to update the table with values for the fields  $total = (paper1 + paper2 + paper3 + paper4 + paper5)$ , result ('passed' if total is greater than or equal to 50% of the total; 'failed' otherwise), and grade ('A' if mark obtained is greater than or equal to 90% of the total mark, 'B' if mark obtained is greater than or equal to 75% of the total mark, 'C' if mark obtained is greater than or equal to 60% of the total mark, 'D' if mark obtained is greater than or equal to 50% of the total mark, and 'F' if mark obtained is less than 50% of the total mark). Display a report in descending order of the total mark, showing the data entered into the table along with the total marks, result and grade.

**Lab 14:** Create a STUDENT table with following fields:

Every day a newspaper vendor gets newspapers in wholesale from a distributor for Rs. 0.60. At the end of the day, the unsold papers are returned to the distributor for Rs. 0.30 rebate per paper. Create a table VENDOR with following specifications:

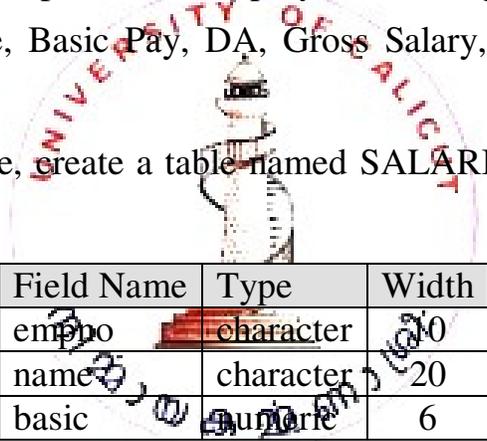
Field	Type
day	date
bought	numeric
sold	numeric
ret	numeric
profit_loss	numeric

Enter the data for day, number of papers bought and number of papers sold for at least 10 days. Write a PL/SQL program process the data to find the number of papers returned and the profit or loss for each day. Display the report showing Date, Number of papers bought, Number of papers sold, Number of papers returned, Profit/Loss and a remark showing whether the days business is a loss, profit, or nil.

**Lab 15:** Prepare a salary report of the employees showing the details such as:

Emp.No, Name, Basic Pay, DA, Gross Salary, PF, Net Salary, Annual Salary and Tax

For this purpose, create a table named SALARIES having the following structure.



Field Name	Type	Width
empno	character	10
name	character	20
basic	numeric	6

Enter the records of at least 10 employees. Use the following information for calculating the details for the report:

DA is fixed as the 40% of the basic pay.

PF is fixed as 10% of the basic pay.

Gross Salary is (Basic Pay + DA).

Net Salary is (Gross Salary – PF)

Annual Salary is (12 \* Net Salary)

Tax is calculated using the following rules:

If annual salary is less than 100000, No Tax

If annual salary is greater than 100000 but less than or equal to 150000, , then the tax is 10% of the excess over 100000.

If annual salary is greater than 150000 but less than or equal to 250000, , then the tax is 20% of the excess over 150000.

If annual salary is greater than 250000, then the tax is 30% of the excess over 250000.

**Lab 16:** Generate a Hospital information system that can generate the following reports:

- Patients who belongs to in-patient category
- Patients who belongs to out-patient category

For this purpose, create a table named HOSPITAL having the following structure.

Field Name	Type	Width
patientid	character	10
name	character	20
age	numeric	3
doctor	character	20
patienttype	character	15
consultcharge	numeric	6
bloodtestcharge	numeric	6
xraycharge	numeric	6
othercharges	numeric	6
totalamount	numeric	6

Enter the records of at least 10 patients. Write a PL/SQL program to display the report in the ascending order of patient name.

**Lab 17:** Using the Hospital table created in Lab 16, generate a Hospital information system that can generate the following reports:

- Patients undergone blood test.
- Patients who have taken X-Rays

**Lab 18:** Design a Hotel Bill calculating system that generates hotel bills for the customers.

**Lab 19:** Design a Hostel Accounting system that generates the Hostel Due Report.

**Lab 20:** Design an Electricity Bill Report generating system that generates electricity bills details of customers for a month.

Lab 21: Generate a Library Information System that generates report of the books available in the library.

Lab 22: Programs involving multiple tables.



## FIFTH SEMESTER

### BCS5B08: Computer Organization and Architecture

**Course Number:** 26

**Contact Hours per Week:** 4 (4T + 0P)

**Number of Credits:** 5

**Number of Contact Hours:** 75 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Aim of the Course

- To learn basic Architecture of a Computer
- To learn basic Computer Organization.

#### Prerequisites

- Basic knowledge of Computer.

#### Course Outline

##### Module I [15T]

Basic Computer Organization and Design: Instruction Codes , Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory reference Instructions, Input, Output and Interrupt Design of Basic Computer, Design of Accumulator logic.

##### Module II [15T]

Micro programmed Control: Control Memory, Address sequencing, Micro program Example, Design of control unit. Processor Organization: general register organization, stack organization, instruction formats, addressing modes, data transfer and manipulation, program control. Computer Arithmetic: Addition, Subtraction, Multiplication, Division algorithms - Floating point arithmetic operations, Decimal arithmetic operations.

### **Module III [15T]**

Memory Organization: Memory Hierarchy, Main memory (RAM/ROM chips), Auxiliary memory, Associative memory, Cache memory, Virtual Memory, Memory Management Hardware, hit/miss ratio, magnetic and optical storage devices

### **Module IV [15T]**

Input-Output Organization: Peripheral devices, I/O interface, Modes of Transfer, Priority Interrupt, Direct Memory Access, Input-Output Processor, and Serial Communication. I/O Controllers, Asynchronous data transfer, Strobe Control, Handshaking.

### **Module V [15T]**

Parallel Processing: Basic Parallel Processing Architecture - Taxonomy- SISD, MISD, SIMD, MIMD structures - CISC vs RISC - Symmetric Multiprocessors - Cache coherence and MESI protocol - Clusters - Non Uniform Memory Access. Pipelining: Basic Concepts of pipelining, Instruction Pipelining. Hazards, Reservation Tables, Collision, Latency, Dynamic pipeline, Vector processing & Vector processors

### **References:**

1. M. Morris Mano, *Computer System Architecture* PHI
2. William Stallings, *Computer Organization and Architecture*, PHI.

## **BCS5B09: Java Programming**

**Course Number:** 27

**Contact Hours per Week:** 6 (3T + 3P)

**Number of Credits:** 4

**Number of Contact Hours:** 90 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To have a review on concept of OOP.
- To learn Java Programming Environments.
- To practice programming in Java.
- To learn GUI Application development in JAVA.

### **Prerequisites:**

- Knowledge in OOP & Programming.

### **Course Outline**

#### **Module I [9T+9L]**

Introduction to Java: History, Versioning, The Java Virtual Machine, Writing a Java Program, Packages, Simple Java Programs. Language Components: Primitive Data Types, Comments, The for Statement, The if Statement, The while and do while Statements, The switch Statement, The break Statement, The continue Statement, Operators - Casts and Conversions, Keywords.

#### **Module II [9T+9L]**

Object-Oriented Programming: Defining New Data Types, Constructors, The String Class, String Literals, Documentation, Packages, The StringBuffer Class, Naming Conventions, The Date Class, The import Statement, Deprecation, The StringTokenizer Class. Methods: Introduction - Method Signatures, Arguments and Parameters, Passing Objects to Methods, Method Overloading, Static Methods, The Math Class, The System Class, Wrapper Classes Arrays: Processing Arrays, Copying Arrays, Passing Arrays to Methods, Arrays of Objects, The Arrays Class, Command Line Arguments, Multidimensional Arrays. Encapsulation: Constructors, The this Reference, Data Hiding, public and private Members, Access Levels, Static Data Members Inheritance & Polymorphism: Inheritance, extends keyword, Polymorphism, The Object Class, Method Overloading & Overriding. Abstract Classes and Interfaces: Abstract Classes, Abstract Class Example, Extending an Abstract Class, Interfaces.

### **Module III [9T+9L]**

Exceptions, I/O and Threads Input and Output in Java: The File Class, Standard Streams, Keyboard Input, File I/O Using Byte Streams, Character Streams, File I/O Using Character Streams - Buffered Streams, File I/O Using a Buffered Stream, Keyboard Input Using a Buffered Stream, Writing Text Files. Threads: Threads vs. Processes, Creating Threads by Extending Thread, Creating Threads by Implementing Runnable, Advantages of Using Threads, Daemon Threads, Thread States, Thread Problems, Synchronization. Exceptions: Exception Handling, The Exception Hierarchy, Triggering Exceptions with throws, Suppressing Exceptions with throw, Developing user defined Exception Classes-The finally Block.

### **Module IV [9T+9L]**

Collections & Database Connectivity Collections: Vectors, Hashtables, Enumerations, Properties, Collection, Framework Hierarchy, Lists, Sets, Maps, The Collections Class. Networking: Networking Fundamentals, The Client/Server Model, InetAddress, URLs, Sockets, Writing Servers, Client/Server Example. Introduction to JDBC : The JDBC Connectivity Model, Database Programming, Connecting to the Database, Creating a SQL Query, Executing SQL Queries, Getting the Results, Updating Database Data, Executing SQL Update/Delete, Error Checking and the SQLException Class, The Statement Interface, The ResultSet Interface, ResultSetMetaData, Transaction Management.

### **Module IV [9T+9L]**

Applets, Events and GUI Applications: Introduction to GUI Applications - Applets - Types of Applet, Applet Skeleton, Update Method, Html Applet tag and passing parameter to applet. Event Handling: The Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter Classes, Inner Classes. Java Desktop Applications, Introduction to the AWT, Overview of the AWT, Structure of the AWT, The AWT hierarchy, Working with:Color, Button, Canvas, Checkbox, Choice, Frame, Label, List, Scroll bar, TextArea, TextField, Font, FontMetrics, Graphics, Image, Menu Component, MenuBar, MenuItem, Checkbox MenuItem,

Menu, Point, Polygon, Rectangle, Layout Manager, Menu Component, Containers, Components, Event handling, Simple Graphics Drawing Lines, Rectangles, etc.

### **References:**

1. Herbert Schildt, *Java Complete Reference*, TMH
2. Jim Keogh, *J2EE Complete Reference*, TMH
3. David Flanagan, Jim Farley, William Crawford, Kris Mangnusson, *Java Enterprise in a Nutshell*, OReill.
4. Patrick N, Schildt H, *Java 2 The Complete Reference*, TMH

## **BCS5B10: Web Programming Using PHP**

**Course Number:** 28

**Contact Hours per Week:** 6 (3T + 3P)

**Number of Credits:** 4

**Number of Contact Hours:** 70 Hrs.

**Course Evaluation:** Internal 20 Marks + External 80 Marks

### **Aim of the Course**

- To learn client side and server side scripting.
- To learn PHP Programming.
- To learn how to develop dynamic websites.
- To learn how to interact with databases through internet.

### **Prerequisites**

- Knowledge in Programming

## **Course Outline**

### **Module I [9T+9L]**

www, W3C, Web Browser, Web server, Web hosting, Web Pages, DNS, URL, Introduction e-documents - Static, Active & Dynamic. Web programming - client-side scripting and server-side scripting. HTML: Introduction to HTML, Basic formatting tags: heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image. Different attributes like align, color, bgcolor, font face, border, size. Navigation Links using anchor tag: internal, external, mail and image links. Lists: ordered, unordered and definition, Table tag, HTML Form controls: form, text, password, textarea, button, checkbox, radio button, select box, hidden controls, Frameset and frames CSS: Introduction to Cascading Style Sheet (CSS), CSS Syntax, Comments, Id and Class, Background - Background Color, Background Image -Text - Text Color, Text Alignment, Text Decoration, Text Transformation, Text Indentation - CSS Font - Font Families, Font Style, Font Size -Setting Text Size - Using Pixels and Em - CSS Lists - Different List Item Markers, Unordered List, Ordered List, An Image as The List Item Marker - CSS Tables - Table Borders, Collapse Borders, Table Width and Height, Table Text Alignment, Table Padding, Table Color CSS Positioning - Static Positioning, Fixed Positioning, Relative Positioning, Absolute Positioning, Overlapping Elements - Float - Horizontal Align - Image Gallery - Image Opacity/Transparency - Image Sprites

### **Module II [9T+9L]**

Javascript: Introduction, Client side programming, script tag, comments, variables, Document Methods: write and writeln methods, alert, Operators: Arithmetic, Assignment, Relational, Logical, Javascript Functions, Conditional Statements, Loops, break and continue. Events Familiarization: onLoad, onClick, onBlur, onSubmit, onChange

### **Module III [9T+9L]**

PHP: Introduction to PHP, Server side scripting, Role of Web Server software, including files, comments, variables and scope, echo and print, Operators: Logical,

Comparison and Conditional operators, Branching statements, Loops, break and continue, PHP functions.

#### **Module IV [9T+9L]**

Working with PHP: Passing information between pages, HTTP GET and POST method, String functions: strlen, strpos, strstr, strcmp, substr, str\_replace, string case, Array constructs: array(),list() and foreach(), PHP advanced functions: Header , Session, Cookie, Object-Oriented Programming using PHP: class, object, constructor, destructor and inheritance.

#### **Module V [9T+9L]**

PHP & MySQL: Features of MySQL, data types, Introduction to SQL commands - SELECT, DELETE, UPDATE, INSERT, PHP functions for MySQL operations: mysql\_connect, mysql\_select\_db, mysql\_query, mysql\_fetch\_row, mysql\_fetch\_array, mysql\_fetch\_object, mysql\_result, Insertion and Deletion of data using PHP, Displaying data from MySQL in webpage. Introduction to AJAX, Implementation of AJAX in PHP, Simple examples like partial page update, Concept of master page, applying templates.

#### **References:**

1. Jon Duckett, *Beginning Web Programming with HTML,XHTML, CSS*, Wrox.
2. Jim Converse & Joyce Park, *PHP & MySQL Bible*, Wiley.
3. Deitel, Harvey M. and Paul J., 3/E, *Internet & World Wide Web How To Program*, 2004
4. Kogent Solutions, *HTML 4.0 in Simple Steps*, Wiley
5. Ed Tittel & Mary Burmeis – Ter, *HTML 4 for Dummies*, Wiley
6. D W Mercer,A Kent,S D Nowicki, *Beginning PHP*, Wrox.
7. Janet Valad, *PHP & MYSQL For Dummies*, Wiley

## **BCS5B11: Principles of Software Engineering**

**Course Number:** 29

**Contact Hours per Week:** 4 (4T + 0P)

**Number of Credits:** 4

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To learn engineering practices in Software development.
- To learn various software development methodologies and practices.
- To learn and study various Evaluation methods in Software Development.

### **Prerequisites**

- Knowledge in Programming.

### **Course Outline**

#### **Module I [12T]**

Introduction to Software Engineering, Software Engineering Approach, Software Process, Characteristics of a Software Process, Software Development Process Models, Process Management Process

#### **Module II [12T]**

Software Requirements, Problem analysis, Requirements Specification, Functional specification with use case, Validation.

#### **Module III [12T]**

Function oriented Design Principles, Module Level Concepts, Structured Design Methodology, Verification, Object oriented analysis and design, Object Oriented Concepts, Design Principles, Unified Modelling Language, Design Methodology

for Object oriented design. Verification- Design walkthrough, Critical Design Review, Consistency checks

#### **Module IV [12T]**

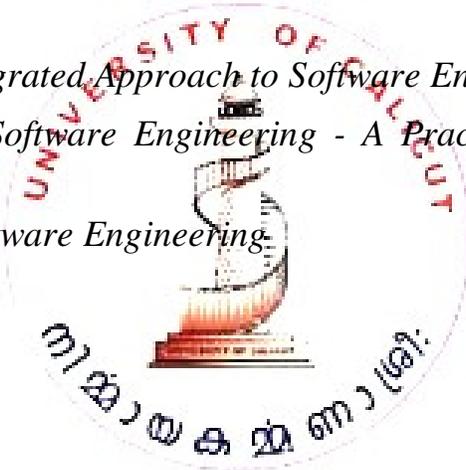
Coding - Programming Principles and Guidelines, Coding Process, Verification Code Inspection, Unit Testing

#### **Module V [12T]**

Testing - Testing Fundamentals, Black Box Testing, White Box Testing, Testing Process

#### **References:**

1. Pankaj Jalote, *An Integrated Approach to Software Engineering*, Narosa Pub.
2. Roger S. Pressman, *Software Engineering - A Practical Approach*, McGraw Hill
3. Ivan Somervelli, *Software Engineering*



## SIXTH SEMESTER

### BCS6B12: Android Programming

**Course Number:** 31

**Contact Hours per Week:** 6 (4T + 1P)

**Number of Credits:** 4

**Number of Contact Hours:** 75 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Aim of the Course

- To have a review on concept of Android programming.
- To learn Android Programming Environments.
- To practice programming in Android.
- To learn GUI Application development in Android platform with XML

#### Prerequisites

- Knowledge in OO & Java Programming

#### Course Outline

##### Module I [12T+3L]

Introducing the android computing platform, History of android, android software stack, Developing end user application using android SDK, android java packages, Setting up the development environment, Installing android development tools (ADT), Fundamental components, Android virtual devices, Running on real device, Structure of android application, Application life cycle.

##### Module II [12T+3L]

Understanding android resources - String resources, Layout resources, Resource reference syntax, Defining own resource IDs - Enumerating key android resources, string arrays, plurals, Colour resources, dimension resources, image resources, Understanding content providers - android built in providers, exploring databases

on emulator, architecture of content providers, structure of android content URIs, reading data using URIs, using android cursor, working with where clause, inserting updates and deletes, implementing content, Understanding intents - basics of intents, available intents, exploring intent composition, Rules for Resolving Intents to Their Components, ACTION PICK, GET CONTENT, pending intents

### **Module III [12T+3L]**

User interfaces development in android - building UI completely in code, UI using XML, UI in XML with code, Android's common controls - Text controls, button controls, checkbox control, radio button controls, image view, date and time controls, map view control, understanding adapters, adapter views, list view, grid view, spinner control, gallery control, styles and themes, Understanding layout managers - linear layout manager, table layout manager, relative layout manager, frame layout manager, grid layout manager.

### **Module IV [12T+3L]**

Android menus - creating menus, working with menu groups, responding to menu items, icon menu, sub menu, context menu, dynamic menus, loading menu through XML, popup menus, Fragments in android - structure of fragment, fragment life cycle, fragment transaction and back stack, fragment manager, saving fragment state, persistence of fragments, communications with fragments, startActivity() and set-TargetFragment(), using dialogs in android, dialog fragments, working with toast, Implementing action bar - tabbed navigation action bar activity, implementing base activity classes, tabbed action bar and tabbed listener, debug text view layout, action bar and menu interaction, list navigation action bar activity, spinner adapter, list listener, list action bar, standard navigation action bar activity, action bar and search view, action bar and fragments.

### **Module V [12T+3L]**

Persisting data - Files, saving state and preferences - saving application data, creating, saving and retrieving shared preferences, preference framework and preference activity, preference layout in XML, native preference controls, preference fragments, preference activity, persisting the application state, including

static files as resources, Working with file system, SQLite - SQLite types, database manipulation using SQLite, SQL and database centric data model for android, android database classes.

### **References:**

1. *Pro Android 4*, Satya Komatineni & Dave MacLean, Apress.
2. *Professional Android 4 Application Development*, Retomeier, Wrox.
3. *Programming Android*, Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura, O'Reilly

## **BCS6B13: Fundamentals of Operating Systems**

**Course Number:** 32

**Contact Hours per Week:** 4 (4T+ 1P)

**Number of Credits:** 4

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To learn objectives & functions of Operating Systems.
- To understand processes and its life cycle.
- To learn and understand various Memory and Scheduling Algorithms.
- To have an overall idea about the latest developments in Operating Systems.

### **Prerequisites**

- Knowledge in Data structures.

### **Course Outline**

#### **Module I [12T+3P]**

Operating System Objectives and functions: The Evolution of Operating Systems, Serial Processing, Simple batch Systems, Multi Programmed batch Systems, Time

Sharing Systems, Parallel Systems, Distributed Systems, Real time systems.

### **Module II [12T+3P]**

Definition of Process, Process States, Process Control Block, Operations on Process, Process Communication, Communication in Client server System, Basic concepts of threads, Concurrency, Principles of Concurrency, Mutual exclusion, Semaphores, Messages, Dead lock, Dead lock Prevention, Dead lock detection, dead lock avoidance.

### **Module III [12T+3P]**

CPU Scheduling: Scheduling Criteria, Scheduling algorithms - FCFS, SJF, Priority, RR, Multilevel, Feedback Queue, Process synchronization, The Critical Section Problem, Synchronization Hardware, Classical Problems of synchronization, File and Database System, File System, Functions of organization, Allocation and Free space management.

### **Module IV [12T+3P]**

Memory Management, Address binding, Logical Vs Physical address space, Dynamic Loading, Dynamic Linking and Shared Libraries, Overlays, Swapping, Contiguous Memory allocation, Paging, Segmentation, Virtual memory, Demand paging, Page replacement, Thrashing.

### **Module V [12T+3P]**

Protection and security: policy and mechanism , authentication, authorization, Mobile OS: Concepts, history, features, architecture, future scope;

Case studies: Android, UNIX kernel and Microsoft windows NT(concept).

### **References:**

1. Silberschatz, Galvin, Gagne, *Operating System Concepts*, John Willey & Sons.
2. Nutt G.J, *Operating Systems: A Modern Perspective*, Addison Wesley.
3. William Stallings, *Operating Systems, Internals and Design Principles*, - PHI.

## **BCS6B14: Computer Networks**

**Course Number:** 33

**Contact Hours per Week:** 5 (5T + 0P)

**Number of Credits:** 4

**Number of Contact Hours:** 70 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To learn about transmissions in Computer Networks.
- To learn various Protocols used in Communication.
- To have a general idea on Network Administration.

### **Prerequisites**

- Knowledge in data structure.
- Knowledge in Operating System.

### **Course Outline:**

#### **Module I [14T]**

Introduction to Computer networks, Topology, categories of networks, Internetwork, Internet, Network Models, Layered model, OSI and TCP/IP models, Physical layer, Switching - Circuit switching, Packet Switching and Message Switching, DTE - DCE Interface, EIA - 232 interface, X.21 modems.

#### **Module II [14T]**

Data link layer, Error detection and correction, Types of errors, Single CSC error and Burst error, Vertical redundancy check (VRC), longitudinal redundancy Check (LRC), Cyclic Redundancy Check (CRC), Error correction - Single CSC error correction, Hamming code Data compression - Huffman code, data link control, Line discipline, Flow control, Error control, Multiple Access, Random Access, ALOHA, pure ALOHA and slotted ALOHA, CSMA/CD and SCMA/CA,

Polling, Wired LANs, Ethernet - IEEE standards, Wireless LANs - IEEE - 802.11, Bluetooth

### **Module III [14T]**

Network layer, Networking and Internetworking devices - Repeaters, Bridges, Routers, Gateways, Logical addressing - IPv4 & IPv6 addresses, Network Address Translation(NAT), Internet protocols, internetworking, Datagram, Transition from IPv4 to IPv6, Address Mapping-Error reporting and multicasting - Delivery, Forwarding and Routing algorithms, Distance Vector Routing, Link State Routing, Multicast routing protocols, The Dijkstra Algorithm.

### **Module IV [14T]**

Transport layer, Process-to-process Delivery:UDP, TCP and SCTP, Congestion control and Quality of Service, Application Layer, Domain Name Systems-Remote Login-Email-FTP, WWW, HTTP; Network management:SNMP, Network security, Cryptography.

### **Module V [14T]**

Network Administration, IP address - Configuring network host - setting hostname - assigning IP address, configuring the Network Interface card, Setup a LAN with more than two systems, Setting up Internet services File Transfer Protocol(FTP), Trivial File Transfer Protocol(TFTP), Simple Mail Transfer Protocol(SMTP) and Post Office Protocol(POP), Setting up Intranet Services, Network File System(NFS), Network Information Service(NIS) and Dynamic Host Configuration Protocol(DHCP), Samba printing and Web server.

### **References:**

1. Behrouz A Forozan, *Introduction to Data Communications & Networking*, TMH
2. Andrew S. Tanenbaum, *Computer Networks*, PHI
3. William Stallings, *Data and Computer Communications*, VIIth Edition, Pearson Education

4. William Stallings, *Cryptography and Network Security, Principles and Practices*, Prentice Hall of India.
5. Steven Graham and Steve Shah, *Linux Administration: A Beginners Guide*, Third Edition, Dreamtech, 2003.

## **BCS6B15: Programming Laboratory III:Java & PHP Programming**

**Course Number:** 34

**Contact Hours per Week:** 2 (0T + 2P)

**Number of Credits:** 2

**Number of Contact Hours:** 30 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To practice Java programming.
- To practice client side and server side scripting.
- To practice PHP Programming.
- To practice developing dynamic websites.
- To practice how to interact with databases through PHP.

### **Prerequisites**

- Theoretical knowledge in Java programming.
- Theoretical knowledge of PHP Programming.

### **Course Outline**

#### **Part A: Java Programming**

**Lab 1:** Data types, variables and operators

**Exercise 1:** Write a program in Java to implement the formula (Area = Height Width) to find the area of a rectangle. Where Height and Width are the rectangles height and width

**Exercise 2:** Write a program in Java to find the result of following expression (Assume a = 10, b = 5)

- i)  $(a \ll 2) + (b \gg 2)$
- ii)  $(a) \parallel (b > 0)$
- iii)  $(a + b * 100) / 10$
- iv)  $a \& b$

**Exercise 3:** Write a program in Java to explain the use of break and continue statements.

**Exercise 4:** Write a program in Java to find the average of marks you obtained in your 10+2 class.

**Lab 2:** Statements and array

**Exercise 1:** Write a program in Java to find AB where A is a matrix of 33 and B is a matrix of 34. Take the values in matrixes A and B from the user.

**Exercise 2:** Write a program in Java to compute the sum of the digits of a given integer. Remember, your integer should not be less than the five digits. (e.g., if input is 23451 then sum of the digits of 23451 will be 15)

**Lab 3:** Class and Objects

**Exercise 1:** Write a program in Java with class Rectangle with the data fields width, length, area and colour. The length, width and area are of double type and colour is string type. The methods are set\_length (), set\_width (), set\_colour(), and find\_area (). Create two object of Rectangle and compare their area and colour. If area and color both are the same for the objects then display Matching Rectangles, otherwise display Non matching Rectangle.

**Exercise 2:** Create a class Account with two overloaded constructors. The first constructor is used for initializing, the name of account holder, the account number and the initial amount in the account. The second constructor is used for initializing the name of the account holder, the account number, the addresses, the type of account and the current balance. The Account class is having methods Deposit (), Withdraw (), and

Get\_Balance(). Make the necessary assumption for data members and return types of the methods. Create objects of Account class and use them.

**Exercise 3:** Write a program in Java to create a stack class of variable size with push() and pop () methods. Create two objects of stack with 10 data items in both. Compare the top elements of both stack and print the comparison result.

#### Lab 4: Inheritance and polymorphism

**Exercise 1:** Write a Java program to show that private member of a super class cannot be accessed from derived classes.

**Exercise 2:** Write a program in Java to create a Player class. Inherit the classes Cricket\_Player, Football-Player and Hockey\_ Player from Player class.

**Exercise 3:** Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days s/he works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program.

**Exercise 4:** Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges.

#### Lab 5: Package and Interface

**Exercise 1:** Write a program to make a package Balance in which has Account class with Display .Balance method in it. Import Balance package in another program to access Display-Balance method of Account class.

**Exercise 2:** Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program.

**Exercise 3:** Create an Interface having two methods division and modules. Create a class, which overrides these methods.

**Exercise 4:** Write a program in Java which implements interface Student which has two methods Display\_Grade and Attendance for

PG\_Students and UG\_Students (PG\_Students and UG\_Students are two different classes for Post Graduate and Under Graduate students respectively).

**Lab 6: Exception Handling**

**Exercise 1:** Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle `ArrayIndexOutOfBoundsException`, so that any such problem doesn't cause illegal termination of program.

**Exercise 2:** Write a Java program to enable the user to handle any chance of divide by zero exception.

**Exercise 3:** Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

**Exercise 4:** On a single track two vehicles are running. As vehicles are going in same direction there is no problem. If the vehicles are running in different direction there is a chance of collision. To avoid collisions write a Java program using exception handling. You are free to make necessary assumptions.

**Lab 7: Multithreading.**

**Exercise 1:** Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting. **Exercise 2:** Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization. **Exercise 3:** Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.

**Exercise 2:** Write a Java program using thread synchronization in multithreading (You can take some objects visible on screen for real time effect).

**Lab 8: Reading, Writing and String handling in Java**

**Exercise 1:** Write a program in Java to create a String object. Initialize this object with your name. Find the length of your name using the appropriate String method. Find whether the character *a* is in your name or not; if yes find the number of times *a* appears in your name. Print locations of occurrences of *a*. Try the same for different String objects.

**Exercise 2:** Write a program in Java for String handling which performs the following:

- a. Checks the capacity of StringBuffer objects.
- b. Reverses the contents of a string given on console and converts the resultant string in upper case.
- c. Reads a string from console and appends it to the resultant string of ii.

**Exercise 3:** Write a program for searching strings for the first occurrence of a character or substring and for the last occurrence of a character or substring.

**Exercise 4:** Write a program in Java to read a statement from console, convert it into upper case and again print on console.

**Exercise 5:** Write a program in Java, which takes the name of a file from user, read the contents of the file and display it on the console.

**Exercise 6:** Write a Java program to copy a file into another file.

#### Lab 9: Applets and its applications

**Exercise 1:** Write a Java Applet program which reads your name and address in different text fields and when a button named find is pressed the sum of the length of characters in name and address is displayed in another text field. Use appropriate colors, layout to make your applet look good.

**Exercise 2:** Create an applet which displays a rectangle/string with specified colour & coordinate passed as parameter from the HTML file.

**Exercise 3:** Create an applet which will display the calendar of a given date.

**Exercise 4:** Write a program to store students detail using Card Layout.

**Exercise 5:** Write a Java Applet program, which provides a text area with horizontal and vertical scrollbars. Type some lines of text in the text area

and use scrollbars for movements in the text area. Read a word in a text field and find whether the word is in the content of the text area or not.

**Lab 10: Networking and other advanced feature and JAVA**

**Exercise 1:** Write a Java program to find the numeric address of the following websites

- www.indiatimes.com
- www.rediff.com
- www.apple.com

In addition to this, find the Internet Address of your local host.

**Exercise 2:** Create an applet which takes name and age as parameters and display the message <name> is <age> year old. Print the URL of the class file.

**Exercise 3:** Write a program to test Socket functionality for appropriate hostname and port number.

**Exercise 4:** Write a Java program to connect to a database created in MySQL/PostgreSQL/MS ACCESS using JDBC concept. Perform basic operations of Selection, Insertion and Deletion on the database.

**Part B: PHP Programming**

**Lab 1: Setting-up the environment**

- Setup WAMP/XAMPP Server or Setup Apache, MySQL and PHP separately in your PHP Lab.
- Simple PHP program that displays a welcome message.
- Write a php program to generate a random number between 1 and 100.
- Modify above program to accept range of the random number from HTML interface.

**Lab 2: Programs involving various control structures like:**

- 'if, else, elseif/else if'
- Alternative Syntax for 'if, else, elseif/else if'

**Lab 3: Programs involving various control structures like:**

- while, do-while, for, foreach, switch, break, continue.
- Try alternative syntax for while, do-while, for, foreach, switch.

Lab 4: Programs involving the following.

- declare, return.
- require, include, require- once, include\_once and goto.

Lab 5: Programs to demonstrate PHP Array functions.

- PHP Array Sorting,
- PHP Key Sorting,
- PHP Value Sorting,
- PHP MultiArray Sorting,
- PHP Array Random Sorting,

Lab 6: Programs to demonstrate PHP Array functions.

- PHP Array Reverse Sorting,
- Array to String Conversion,
- Implode() function,
- String to Array, ArrayCount,
- Remove Duplicate Values

Lab 7: Programs to demonstrate PHP Array functions.

- Array Search,
- Array Replace,
- Array Replace Recursive,
- Array Sub String Search

Lab 7: Demonstrate the following.

- Use of regular expression to compare two strings.
- Extract Domain name from URL.
- Find the number of rows from a mysql database for your query.

Lab 8: Generate a Guestbook which will allow your website visitor to enter some simple data about your website.

Lab 9: Develop a PHP program for Email Registration.

Lab10: Develop a project for making Application form and performing Degree Admission On-line.

## **BCS6B16: Programming Laboratory IV: Android & Linux shell Programming**

**Course Number:** 35

**Contact Hours per Week:** 2 (0T + 2P)

**Number of Credits:** 2

**Number of Contact Hours:** 30 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To practice Android programming.
- To practice user interface applications.
- To develop mobile application.
- To practice shell programming

### **Prerequisites:**

- Theoretical knowledge in Android programming.
- Theoretical knowledge of Shell Programming.

### **Course Outline:**

#### **Part A: Android Programming**

Lab 1: Programs to understand basic arithmetic operations

Lab 2: Programs to understand basic logic operations

Lab 3: Programs to understand loops and control statements

Lab 4: Programs to understand GUI in android

Lab 5: Android application for adding two numbers

Lab 6: Develop simple user interface to display message

Lab 7: Create two menu items-opening a file-saving a file

Lab 8: Text view controls to represent each row in a list view

Lab 9: Implementation of background image

- Lab 10: Starting another activity from your own activity using intent
- Lab 11: Create a new activity that services ACTION-PICK for contact data which display each of the contact in the contact database and lets the user to select one before closing and returning the selected contacts URL to the calling activities
- Lab 12: Create Android application to linkify a text view to display web and E-mail address as hyperlinks. When clicked they will open the browser and E-mail address respectively
- Lab 13: Implementation of array adapter
- Lab 14: Create an alert dialogs used to display a message and offer two button options to continue. Clicking either button will close the dialog after executing the attached click listener
- Lab 15: Create an earth quake viewer
- Lab 16: Create mobile applications
- Lab 17: Program to implement simple calculator
- Lab 18: Program to Get IP Address
- Lab 19: Program to Home And Lock Screen Widget (Temperature Widget)
- Lab 20: Program to Device/Battery Temperature Sensor
- Lab 21: Program to Audio Demo ~~AudioTrack~~, ~~AudioRecord~~
- Lab 22: Program to Blocking Incoming call Android
- Lab 23: Program to create simple login screen.

### **Part B: Shell Programming**

- Lab 1: Implementation of file and directory operation commands
- Lab 2: Implementation of filters and pipes
- Lab 3: Simple shell programs including arithmetic operators
- Lab 4: Shell programs including if statement and case statement
- Lab 5: Shell programs including for, while, until loop
- Lab 6: Shell programs for reading and writing files
- Lab 7: Implementing string operations

Lab 8: Implementing communication commands

Lab 9: Implementation of managing user accounts

Lab10: Implementing backup and restore

Lab 11: Configuration of web servers and proxy server

Lab 12: Configuration of DNS servers and mail server

## **BCS6B18: Project**

**Course Number:** 37

**Contact Hours per Week:** 4 (0T + 2P in V Sem + 2 P in VI Sem)

**Number of Credits:** 2

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 10 Marks + External – 40 Marks

### **Aim of the Course**

- To provide practical knowledge on software development process

### **Prerequisites**

- Basic programming and system development knowledge

### **Course Outline**

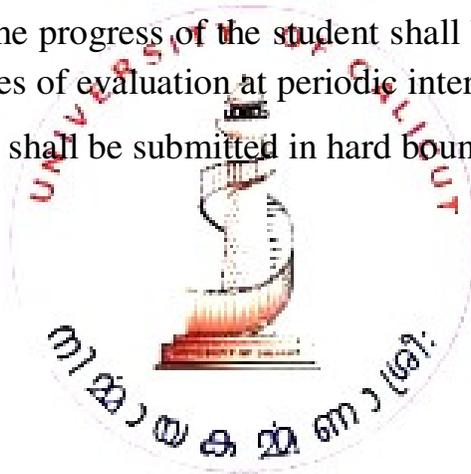
The objective of the B. Sc Computer Science final project work is to develop a quality software solution by following the software engineering principles and practices. During the development of the project the students should involve in all the stages of the software development life cycle (SDLC). The main objective of this project course is to provide learners a platform to demonstrate their practical and theoretical skills gained during five semesters of study in B. Sc Computer Science Programme. During project development students are expected to define a project problem, do requirements analysis, systems design, software development, apply testing strategies and do documentation with an overall emphasis on the development of a robust, efficient and reliable software systems. The project

development process has to be consistent and should follow standard. For example database tables designed in the system should match with the E-R Diagram. SRS documents to be created as per IEEE standards.

Students are encouraged to work on a project preferably on a live software project sponsored by industry or any research organization. Topics selected should be complex and large enough to justify as a B.Sc Computer Science final semester project. The courses studied by the students during the B. Sc Computer Science Programme provide them the comprehensive background knowledge on diverse subject areas in computer science such as computer programming, data structure, DBMS, Computer Organization, Software Engineering, Computer Networks, etc., which will be helping students in doing project work. Students can also undertake group project to learn how to work in groups.

For internal evaluation, the progress of the student shall be systematically assessed through two or three stages of evaluation at periodic intervals.

A bonafied project report shall be submitted in hard bound complete in all aspects.



## OPEN COURSES

### **BCS5D01: Introduction to Computers & Office Automation**

**Course Number:** XX

**Contact Hours per Week:** 2 (2T + 0P)

**Number of Credits:** 2

**Number of Contact Hours:** 30 Hrs.

**Course Evaluation:** Internal – 10 Marks + External – 40 Marks

#### **Aim of the Course**

- To learn Office Automation.

#### **Prerequisites**

- Basic knowledge in Computer & Internet.

#### **Course Outline**

##### **Module I [7T]**

Introduction to Computers: Types of Computers - DeskTop, Laptop, Notebook and Netbook. Hardware: CPU, Input / Output Devices, Storage Devices – System - Software - Operating Systems, Programming Languages, Application Software - Networks - LAN, WAN - Client - Server.

##### **Module II [7T]**

Documentation Using a Word Processor (OpenOffice Writer / M.S. Word) - Introduction to Office Automation, Creating & Editing Document, Formatting Document, Auto-text, Autocorrect, Spelling and Grammar Tool, Document Dictionary, Page Formatting, Bookmark, Advance Features - Mail Merge, Macros, Tables, File Management, Printing, Styles, linking and embedding object, Template.

### **Module III [8T]**

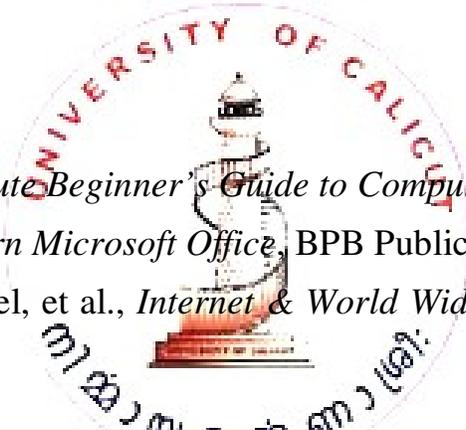
Electronic Spread Sheet (OpenOffice Calc/MS-Excel) - Introduction to Spread Sheet, Creating & Editing Worksheet, Formatting and Essential Operations, Formulas and Functions, Charts, Advanced features - Pivot table & Pivot Chart, Linking and Consolidation.

### **Module IV [8T]**

Presentation using (OpenOffice Impress/MS-Power Point): Presentations, Creating, Manipulating & Enhancing Slides, Organizational Charts, Charts, Word Art, Layering art Objects, Animations and Sounds, Inserting Animated Pictures or Accessing through Object, Inserting Recorded Sound Effect or In-Built Sound Effect.

### **References:**

1. Michael Miller, *Absolute Beginner's Guide to Computer Basics*, Prentice Hall.
2. Russell A. Stultz, *Learn Microsoft Office*, BPB Publication.
3. H.M.Deitel, P. J. Deitel, et al., *Internet & World Wide Web - How to program*, Prentice Hall.



## **BCS5D02: Web Designing**

**Course Number:** XX

**Contact Hours per Week:** 2 (2T + 0P)

**Number of Credits:** 2

**Number of Contact Hours:** 30 Hrs.

**Course Evaluation:** Internal – 10 Marks + External – 40 Marks

### **Aim of the Course**

- To learn Web designing.

## **Prerequisites**

- Basic knowledge in Computer & Internet.

## **Course Outline**

### **Module I [7T]**

HTML: Introduction - history of html, sgml - structure of html document, web page layout, html tags and types - font type, paragraph formatting, meta data, blockquote, hyperlinks, linking, comments, white space, horizontal ruler, images, ordered and unordered lists, frames, tables, forms

### **Module II [7T]**

DHTML: Introduction, DHTML technologies, elements of DHTML, document object model, events - window events, form events, keyboard events, mouse events, style sheets, properties used in style sheets - background properties, positioning properties.

### **Module III [8T]**

Javascript: Introduction and advantages of javascript, java script syntax, writing javascript in html, javascript operators, arrays and expressions, programming constructs - for .. in loop, while loop - dialog boxes and prompts - alert, prompt, confirm methods - functions - built-in functions and userdefined functions, scope of variables, handling events, using event handlers and event methods, form object, properties, methods, form element's properties and methods.

### **Module IV [8T]**

HTML Editor: Introduction, advantages, creating, opening, saving a web page, building forms, formatting and aligning text and paragraph, adding lists, styles and themes, linking pages, working with images, frames.

**Reference:**

1. H. M. Dietel, *Internet and World Wide Web*, Pearson.

**BCS5D02: Introduction to Problem Solving and C Programming**

**Course Number:** XX

**Contact Hours per Week:** 2 (2T + 0P)

**Number of Credits:** 2

**Number of Contact Hours:** 30 Hrs.

**Course Evaluation:** Internal – 10 Marks + External – 40 Marks

**Aim of the Course**

- To introduce fundamental principles of Problem Solving aspects.
- To learn the concept of programming.
- To learn C language.

**Prerequisites**

- Background of the basic science at +2 level

**Course Outline**

**Module I [7T]**

Introduction: The problem solving aspect, Top-down design, Implementation of algorithms, Program verification, efficiency of algorithms. Introduction to C Programming, overview and importance of C, C Program Structure and Simple programs, Creation and Compilation of C Programs under Linux and Windows Platforms.

**Module II [7T]**

Elements of C Language and Program constructs. Character Set, C Tokens, Keywords and Identifier, Constants, Variables, Data types, Variable declaration

and assignment of values, Symbolic constant definition. C-Operators, Arithmetic operators, relational operators, and logical operators, assignment operators, increment and decrement operators, conditional operators, special operators, arithmetic expressions, evaluation of expressions, precedence of arithmetic operators, Type conversion in expressions, operator precedence and associativity, Mathematical Functions, I/O operations.

### **Module III [8T]**

Decision making, Branching and Looping. Decision making with IF statement, Simple IF statement, If.. .else statement, Nesting of If.. .else and else...if Ladder, Switch statement, Conditional operator, Go-to statement. Looping: While loop, Do-While, and For Loops, Nesting of loops, jumps in loop, skipping of loops.

### **Module IV [8T]**

Array & Strings - One dimensional array, two dimensional array and multi-dimensional array, strings and string manipulation functions. Structures & Union structure definition , giving values to members, structure initialization, comparison of structure variables, arrays of structures, arrays within structures, structures within arrays, structures and Functions, Unions, CSC fields.

### **References:**

1. E. Balaguruswami, *Programming in ANSI C*
2. Brian W. Kernighan & Dennis M. Ritchie, *The C Programming Language*
3. Yashvant P. Kanetkar, *Let Us C*
4. Byran Gotfried, *Programming with C*, Schaums Outline Series

## ELECTIVE COURSES

### BCS6B17a: Computer Graphics

**Course Number:** 37

**Contact Hours per Week:** 4 (4T + 0P)

**Number of Credits:** 3

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

#### Aim of the Course

- To learn basics of Computer Graphics.

#### Prerequisites

- Basic knowledge in Mathematics.
- Basic knowledge in Computer.

#### Course Outline

##### Module I [12T]

Introduction to Computer Graphics Definition, Application, Pixel, Frame Buffer, Raster and Random Scan display, Display devices CRT, Color CRT Monitors, basics of LCD & LED Monitors.

##### Module II [12T]

Scan Conversion of line DDA algorithm of line drawing, Scan conversion of circles Bresenham's circle generating algorithm, Polygon Filling Scan line polygon filling algorithm.

##### Module III [12T]

Two Dimensional transformation, Translation, Rotation, Scaling, Homogeneous Coordinates, Reflection, Shear.

### **Module IV [12T]**

Window to view port transformation, clipping, line clipping, Cohen Sutherland line clipping, Polygon clipping, Sutherland and Gary Hodgman polygon clipping algorithm.

### **Module V [12T]**

Color Models & Color Applications Light and Color, Different color models, RGB, CMY, YIQ. Introduction to GIMP Image Manipulation using GIMP.

### **References:**

1. Donald Hearn and M. Pauline Baker, *Computer Graphics*, PHI, New Delhi.
2. Zhigang Xiang and Roy Plasock , *Computer Graphics*, Schaum's Outlines.
3. Deborah Morley, *Understanding Computers Today and Tomorrow*, Introductory Edition.

## **BCS6B17b: System Software**

**Course Number:** 37

**Contact Hours per Week:** 4 (4T + 0P)

**Number of Credits:** 3

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To build fundamental knowledge in system software.
- To learn functions of various system software.
- To learn specifically learn compilation process of a program.

### **Prerequisites**

- Basic knowledge in Programming.

**Course Outline:**

**Module I [12T]**

System software: General concept, Assemblers, loaders, linkers, macros, compilers, interpreters, operating system, Design of assemblers.

**Module II [12T]**

Macros and macro processors, Macro definitions and instructions, Macro calls, Features of Macros, Design of Macroprocessors

**Module III [12T]**

Loading, linking and relocating Loader schemes- Binders, linking loaders, overlays, dynamic binders-Dynamic loading and dynamic linking - Relocatability of programs.

**Module IV [12T]**

Compilers - Phases of a compiler - Lexical syntax, intermediate code generation, optimization, code generation, symbol table and error correcting routines - Passes of a compiler.

**Module V [12T]**

Case studies of lexical and syntax analyzers: LEX and YACC.

**References:**

1. D.M.Dhamdhere, *Systems Programming and Operating Systems*
2. John J Donovan, *Systems programming*

## **BCS6B17c: Microprocessor & Applications**

**Course Number:** 37

**Contact Hours per Week:** 4 (4T + 0P)

**Number of Credits:** 3

**Number of Contact Hours:** 60 Hrs.

**Course Evaluation:** Internal – 20 Marks + External – 80 Marks

### **Aim of the Course**

- To understand internals of Microprocessor.
- To learn architecture of 8086 Microprocessor
- To learn instruction set of 8086 Microprocessor
- To learn how to program a Microprocessor

### **Prerequisites**

- Basic knowledge of Computer.

### **Course Outline**

#### **Module I [12 Hours]**

16-Bit Microprocessor, 8086, Architecture, Pin Configuration, 8086 Minimum and Maximum mode configurations,

#### **Module II [12 Hours]**

Addressing modes, 8086 Instruction set (Data transfer, Arithmetic, Branch, Processor control & String instruction), 8086 interrupts.

#### **Module III [12 Hours]**

Assembler Directives Data Definition And Storage Allocation, Program Organization, Alignment, Program End value, Returning Attribute, Procedure Definition, Macro Definition, Data Control, Branch Displacement, Header File Inclusion, Target Machine Code Generation, Control Directives.

**Module IV [12 Hours]**

Peripherals and Interfacing, Interfacing output displays (8212), interfacing input keyboards, key Debounce, Programmable communication interface (8251A), programmable peripheral interface (8255), Programmable DMA Controller (8257), Programmable interrupt controller (8259), Programmable interval timer (8253).

**Module V [12 Hours]**

Advanced Microprocessors Introduction to 80186, 80286, 80386, 80486 and Pentium processors, General introduction to BIOS and DOS interrupts.

References

1. Barry.B.Brey, *The Intel Microprocessor 8086/8088. 80186, 80286, 80386 and 80486 Architecture Programming and Interfacing*, Prentice Hall of India Pvt.Ltd.1995.
2. K.R. Venugopal, Raj Kumar, *Microprocessor, X86 programming*, BPB publications, New Delhi
3. Abel P, *IBM PC Assembly Language & Programming*, Parson Education Asia
4. B Ram, *Fundamentals of Microprocessors and Microcomputers*, Dhanpat Rai Publications Pvt. Ltd., New Delhi
5. Mohamad Rafiquzzaman, *Microprocessors and Microcomputer Based System Designing*, Universal Bookstall, New Delhi
6. Yu. Cheng Liu, Glenn A Gibson, *Microcomputer Systems The 8086/8088 Family. Architecture, Programming & Designing*, Prentice Hall of India Pvt. Ltd., New Delhi
7. Ray A.K.Bhurchandi.K.M, *Advanced Microprocessor and Peripherals*, Tata McGrawHill.

**UNIVERSITY OF CALICUT**  
**THENHIPALAM, CALICUT UNIVERSITY P.O**



**SYLLABUS**

FOR

**COMPUTER SCIENCE**  
**(COMPLEMENTARY)**  
**(CHOICE BASED CREDIT AND SEMESTER SYSTEM)**

UNDER THE

**FACULTY OF SCIENCE**

**FOR THE STUDENTS ADMITTED FROM THE ACADEMIC**  
**YEAR 2014 – 15 ONWARDS**

**BOARD OF STUDIES IN COMPUTER SCIENCE (UG)**  
**THENHIPALAM, CALICUT UNIVERSITY P.O**  
**KERALA, 673 635, INDIA**

**JULY, 2014**

© COPYRIGHT BY UNIVERSITY OF CALICUT, 2014

**REGULATIONS**  
FOR  
**COMPUTER SCIENCE - COMPLEMENTARY**  
(CHOICE BASED CREDIT AND SEMESTER SYSTEM)

**EFFECTIVE FROM THE ACADEMIC YEAR 2014-15**

## **COURSE EVALUATION**

Total marks for each complementary course, including lab course, shall be 80 marks.

The evaluation scheme for each course shall contain two parts (1) Internal evaluation (2) external evaluation

16 marks shall be given to the internal evaluation. The remaining 64 marks shall be for the external evaluation.

### **INTERNAL EVALUATION**

16 marks in each course, including lab, are for internal examinations.

The internal assessment shall be based on a predetermined transparent system involving written test, assignments, seminars and attendance in respect of theory courses and on test/record/viva/attendance in respect of lab courses.

Components with percentage of marks of Internal Evaluation of Theory Courses are:

Test paper – 8 Marks

Attendance – 4 Marks

Assignment/Seminar/Viva – 4 Marks

Components with percentage of marks of Internal Evaluation of Lab Courses are:

Test paper – 8 Marks

Attendance – 4 Marks

Assignment/Lab involvement – 4 Marks

Attendance of each course will be evaluated as below:

Above 90% attendance – 4 Marks

85 to 89% – 3 Marks

80 to 84% – 2 Marks

76 to 79 % – 1.5 Marks

75% – 1 Marks

(If a fraction appears in the final internal marks awarded, nearest whole number is to be taken)

To ensure transparency of the evaluation process, the internal assessment marks awarded to the students in each course in a semester shall be notified on the notice board at least one week before the commencement of external examination. There shall not be any chance for improvement for internal marks. The course teacher(s) shall maintain the academic record of each student registered for the course, which shall be forwarded to the University by the college Principal after obtaining the signature of both course teacher and HOD.

The marks secured for internal examination only need be sent to university, by the colleges concerned.

**EXTERNAL EVALUATION:**

There shall be University examinations for each course at the end of each semester.

Practical examinations shall be conducted by the University at the end of fourth semester.

External evaluation carries 64 Marks, for each course.

External evaluation of even (2 and 4) semesters will be conducted in centralized valuation camps immediately after the examination. Answer scripts of odd semester (1 and 3) examination will be evaluated by home valuation. All question papers shall be set by the university.

The model of the question paper for external examination (theory courses) of 3 Hrs duration shall be:

1. **Section A:** 9 compulsory objective type questions (MCQ/fill in the blank/ matching/one word/etc) of 1 mark each (**Total 9 Marks**)
2. **Section B:** 5 compulsory short answer type questions of 2 Marks each (either a single question or can have subdivisions) (**Total 10 Marks**)
3. **Section C:** 5 short essay type questions of 5 Marks each, to be attempted from a set of 8 questions – at least one question from each unit (either a single question or can have subdivisions) (**Total 25 Marks**)
4. **Section D:** 2 long essay type questions of 10 Marks each, to be attempted from a set of 3 questions (either a single question or can have subdivisions) (**Total 20 Marks**)

The external examination in theory courses is to be conducted with question papers set by external experts. The evaluation of the answer scripts shall be done by examiners based on a well-defined scheme of valuation and answer keys shall be provided by the University.

The external examination in practical courses shall be conducted by two examiners, one internal and an external, appointed by the University.

The model of the question paper for external examination (lab courses) of 3 Hrs duration shall be:

1. **Section A:** One marked question of 25 Marks from Programming Lab Part A (C Programming) is to be attempted (Design - Algorithm/Flowchart/Interface: 5 Marks, Code: 10 Marks and Result: 10 Marks. **Total 25 Marks**)
2. **Section B:** One marked question of 25 Marks from Programming Lab Part B (VB.NET Programming) is to be attempted (Design - Algorithm/Flowchart/Interface: 5 Marks, Code: 10 Marks and Result: 10 Marks. **Total 25 Marks**)
3. **Section C:** Lab viva voce (**Total 10 Marks**)

#### 4. **Section D:** Lab Record (**Total 4 Marks**)

##### **REVALUATION:**

In the new system of grading, revaluation is permissible. The prevailing rules for revaluation are applicable.

Students can apply for photocopies of answer scripts of external examinations. Applications for photocopies/scrutiny/revaluation should be submitted within 10 days of publication of results. The fee for this shall be as decided by the university.

##### **IMPROVEMENT COURSE**

A maximum of two courses (Common, Core, Complementary or Open) can be improved in each semester. Improvement of a particular semester can be done only once. The student shall avail the improvement chance in the succeeding year after the successful completion of the semester concerned. The internal marks already obtained will be carried forward to determine the grades/marks in the improvement examination. If the candidate fails to appear for the improvement examination after registration, or if there is no change in the results of the improvement examination appeared, the marks/grades obtained in the first appearance will be retained.

Improvement and supplementary examinations cannot be done simultaneously.

##### **EVALUATION AND GRADING**

Mark system is followed instead of direct grading for each question (for both internal and external examinations). For each course in the semester letter grade, grade point and % of marks are introduced in 7- point indirect grading system. The grading on the basis of a total internal and external mark will be indicated for each course and for each semester and for the entire programme.

Indirect Grading System in 7 point scale is as below:

% of Marks	Grade	Interpretation	Grade Point (G)	Range of Grade Points	Class
90 and above	A+	Outstanding	6	5.5 to 6	First class with
80 to below 90	A	Excellent	5	4.5 to 5.49	

					distinction
70 to below 80	B	Very Good	4	3.5 to 4.49	First class
60 to below 70	C	Good	3	2.5 to 3.49	
50 to below 60	D	Satisfactory	2	1.5 to 2.49	Second class
40 to below 50	E	Pass/Adequate	1	0.5 to 1.49	Pass
Below 40	F	Failure	0	0 to 0.49	Fail

An aggregate of E grade with 40% marks (after external and internal put together) is required in each course for a pass and also for awarding a degree.

Appearance for Internal Assessment (IA) and End Semester Evaluation (ESE-external) are compulsory and no grade shall be awarded to a candidate if she/he is absent for IA/ESE or both. For a pass in each course 40% marks or E grade is necessary

A student who fails to secure a minimum grade for a pass in a course is permitted to write the examination along with the next batch.

After the successful completion of a semester, Semester Grade Point Average (SGPA) of a student in that semester is calculated using the formula given below. For the successful completion of a semester, a student should pass all courses. However, a student is permitted to move to the next semester irrespective of SGPA obtained.

The Semester Grade Point Average can be calculated as

$$SGPA = \frac{\text{Sum of the credit points of all courses in a semester}}{\text{Total Credits in that semester}}$$

$$i.e., \quad SGPA = \frac{C1 \times G1 + C2 \times G2 + C3 \times G3 + \dots}{n}$$

where  $G1, G2, \dots$  are grade points of different courses;  $C1, C2, \dots$  are credits of different courses of the same semester and  $n$  is the total credits in that semester.

The Cumulative Grade Point Average (CGPA) of the student is calculated at the end of a programme. The CGPA of a student determines the overall academic level of the student in a programme and is the criterion for ranking the students. CGPA can be calculated by the following formula

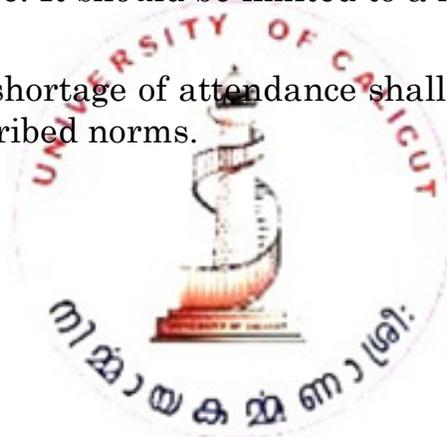
The Cumulative Grade Point Average (CGPA) can be calculated as:

$$CGPA = \frac{\text{Total credit points obtained in six semesters}}{\text{Total Credits}}$$

## ATTENDANCE

A student shall be permitted to appear for the semester examination, only if he/she secures not less than 75% attendance in each semester. Attendance shall be maintained by the concerned Department. Condonation of shortage of attendance to a maximum of 9 days in a semester subject to a maximum of two times during the whole period of the degree programme may be granted by the University. Benefits of attendance may be granted to students who attend the approved activities of college/university with prior concurrence of the Head of the institution. Participation in such activities may be treated as presence in lieu of their absence on production of participation/attendance certificate. It should be limited to a maximum of 9 days in a semester.

The condonation of shortage of attendance shall be granted according to the existing prescribed norms.



**CURRICULUM FOR B.SC COMPUTER SCIENCE  
(COMPLEMENTARY)  
(2014 – 15 ACADEMIC YEAR ONWARDS – AS PER THE  
CUCBCSSUG 2014 REGULATIONS)**

**Total Courses: 5**

**Total Credits: 12**

Semester	Course No	Course Code	Course Title	Marks			Contact Hours			Credits
				Internal	External	Total	Theory	Lab	Total	
I	1	CSC1C01	Computer Fundamentals	16	64	80	2	2	4	3
II	2	CSC2C02	Programming in C	16	64	80	2	2	4	2
III	3	CSC3C03	Fundamentals of System Software, Networks & DBMS	16	64	80	3	2	5	3
IV	4	CSC4C04	Visual Programming	16	64	80	3	0	3	2
IV	5	CSC4C05	Programming Lab: C & Visual Programming	16	64	80	0	2	2	2
<b>Total (5 Courses)</b>						<b>400</b>				<b>12</b>

**CSC1B01 – Computer Fundamentals**

**Semester: 1**

**Course Number: 1**

**Contact Hours: 2T+2L**

**Number of Credits: 2**

**Number of Contact Hours: 30T+30L**

**Course Evaluation: Internal – 15 Marks + External – 65 Marks**

**Aim of the Course:** To impart the students with fundamental principles and operations of various units of computer and to impart them with the basic skill in application packages.

**Objectives of the Course:**

- ✚ To learn the basics of computer hardware units and how they work together
- ✚ To acquire basic skill with office packages

**Prerequisites:** Background of the basic science at +2 level

**Course Outline**

**UNIT I (6T+6L)**

Number systems- Non-positional number systems and positional number systems (Binary, Octal and Hexadecimal), Converting from one number system to another- decimal to a new base, converting to decimal from another bases, converting from base other than ten to base other than ten, short cut method for converting from binary to octal, octal to binary, binary to hexadecimal and hexadecimal to binary, Computer Codes (BCD, EBCDIC, ASCII) error detecting and correcting codes, parity bit, Hamming Code, computer arithmetic ,importance of binary, binary addition and subtraction.

**UNIT II (6T+6L)**

Boolean Algebra and Logic circuits- fundamental concepts of Boolean Algebra, postulates, Principle of duality, theorems of Boolean Algebra, Boolean functions, minimization, complement, canonicals forms, conversion between canonical forms. Logic Gates- AND, OR, NOT, NAND, NOR, XOR and XNOR, logic circuits, converting expression to logic circuit, universal NAND and NOR gates, Exclusive OR and equivalence functions, Design of Combinational circuits (Half Adder, Subtractor and Full Adder)

**UNIT III (6T+6L)**

Basic Computer Organization-Input Unit, Output Unit, Storage Unit (Direct, Sequential and Random Access), CPU organization, Control Unit (micro programmed and hardwired control), primary storage, memory hierarchy, storage locations and addresses, storage capacity, bit, byte, nibble, RAM, ROM, PROM and EPROM, cache memory,

registers. Secondary storage devices (Magnetic tape, Hard disk and CD drive)

#### **UNIT IV (6T+6L)**

I/O devices - Input Devices-identification and its use, keyboard, pointing devices (mouse, touch pad and track ball), Video digitizer, remote control, joystick, magnetic stripes, scanner, digital camera, microphone, sensor, and MIDI instruments, Output Devices-identification and its use, monitor, printer (laser, inkjet, dot-matrix), plotter, speaker, control devices (lights, buzzers, robotic arms, and motors)

#### **UNIT V (6T+6L)**

Planning a Computer program, purpose of program planning, algorithm, flowchart - symbols, sample flowcharts, advantages and limitations

#### **References:**

1. *Computer Fundamentals*, Pradeep K. Sinha and Priti Sinha, BPB
2. *Introduction to Computer*, Peter Nortorn, TMH
3. *Fundamental of Computers*, Rajaraman, V., Prentice Hall India
4. *Computer Fundamentals*, B. Ram

#### **Lab List**

#### **Word Processing**

- ✚ Paragraph formatting
- ✚ Newspaper style Document
- ✚ Table creation
- ✚ Mail merge
- ✚ Page formatting & printing

#### **Spreadsheet**

- ✚ Worksheet entries, including formulas
- ✚ Formatting cells
- ✚ Chart creation

✚ Functions

### **Presentation Software**

✚ Creating presentation

✚ Animations

✚ Sound

✚ Inserting picture

## **CSC2C02 – Programming in C**

**Semester:** 2

**Course Number:** 2

**Contact Hours:** 2T+2L

**Number of Credits:** 2

**Number of Contact Hours:** 30T+30L

**Course Evaluation:** Internal – 15 Marks + External – 65 Marks

**Aim of the Course:** To equip the students with the basic concepts of problem solving using computers.

**Objectives of the Course:**

- ✚ To learn the concepts of programming.
- ✚ To learn the C language

**Prerequisites:** Background of the basic science at +2 level

### **Course Outline**

#### **UNIT I (6T+6L)**

Introduction to C- Structure of C program, Character Set, Keywords, Identifiers, Data Types, Qualifiers, Variables, Declarations, Symbolic Constants, Expressions, Statements, Different Types of Operators (Arithmetic, Logical, Relational & Equality, Unary and Conditional), Operator Precedence and Associativity, Library Functions, Comments, I/O functions-( Formatted scanf() & printf(), getchar (),

putchar (), getche(), gets(), puts())

## **UNIT II (6T+6L)**

Control Statements- Selection Statements (if, if-else, else if ladder, switch), iteration (while, do while, for), jumping (goto, break, continue), Nested Control Statements

## **UNIT III (6T+6L)**

Structured Data types - Arrays (One dimensional and Two Dimensional), Character and String Functions, Structure (Definition, Processing-period Operator), Union

## **UNIT IV (6T+6L)**

User defined Functions - Advantages, Definition, Accessing functions, formal and Actual Parameters, Recursion, Storage Classes- Automatic, External, Static and Register Variable, Argument Passing Mechanism

## **UNIT V (6T+6L)**

Pointers and data files- Pointers, advantages, declaration, operations on pointers, pointers and one dimensional arrays, dynamic memory allocation. Data files (sequential), file handling functions (fopen(), fclose(), fputc(), fgetc(), fgets(), fputs(), fscanf(), fprintf())

## **References:**

1. *Programming in Ansi C*, E Balagurusamy, Tata McGraw Hill
2. *Programming with C*, Byran Gotfried, Schaum Series
3. *Programming in C*, Kezningham & Ritchie
4. *Let us C*, Yashvant Kanetkar, BPB publications
5. *The spirit of C*, Mullish Cooper, Jasco books
6. *The Complete reference C*, Herbert Schildt, Tata Mc Graw Hill

## **C Programming Lab List**

- ✚ Programs involving different data types
- ✚ Programs involving different arrays

- ✚ Programs involving different control structures and nested control structures
- ✚ Programs involving functions
- ✚ Programs involving recursion
- ✚ Programs involving different storage classes
- ✚ Programs involving strings
- ✚ Programs involving structure and union
- ✚ Programs involving files

## CSC3C03 – Fundamentals of System Software, Networks & DBMS

**Semester:** 3

**Course Number:** 3

**Contact Hours:** 3T+2L

**Number of Credits:** 3

**Number of Contact Hours:** 45T+30L

**Course Evaluation:** Internal – 15 Marks + External – 65 Marks

**Aim of the Course:** To impart the students with the basic concepts of system software, Computer Networks and Database.

### Objectives of the Course:

- ✚ To learn the basic concepts of various system software
- ✚ To learn the basics of Computer Networks
- ✚ To learn the basics of Databases

**Prerequisites:** Background of the basic science at +2 level

### Course Outline

#### UNIT I (9T+6L)

System software - classification of programming languages (Machine, assembly & High level), Characteristics and Comparison, language

processors (Assembler, Interpreter and Compiler), Operating Systems- Functions, types of OS (batch, multiprogramming, time sharing, real time and distributed)

### **UNIT II (9T+6L)**

Computer networks- goals of networking, network topologies, types of networks (LAN, MAN and WAN), network model, OSI model- 7 layers, Internet Layer- 5 layers, Communication Media-Guided (Twisted Pair, Coaxial Cable and Fiber Optic) and Unguided (microwave, satellite)

### **UNIT III (9T+6L)**

Database Management Systems-definition, structure of Database, data models (Record based Data model, Network model: - Basic Components, Record types, data types, links, relationships, Hierarchical model and Relational model)

### **UNIT IV (9T+6L)**

Structured query language - Create, insert, select, update, delete, alter, drop commands

### **UNIT V (9T+6L)**

HTML-hypertext, hyper media, understanding basic HTML tools- HTML editor, web browser, General structure of HTML document, different types of elements-doc type, comment element, structural element, HTML tags and attributes: <html>, <body>, <head>, <title>, <h1>,... ,<h6>, <br>, <table>, <img>, <hr>, adding links, background image to the body, creating lists.

### **References:**

1. *Fundamentals of Computers*, P. K Sinha
2. *Operating System: A concept based Approach*, D. M Dhamdhare
3. *Data Communication & Networking*, Behrouz A Forouzan, MC Graw Hill
4. *Principles of Web Page Design*, Joel Sklar, Vikas Publications

## Lab List

### HTML

- ✚ Simple HTML document creation
- ✚ HTML document with tables
- ✚ HTML document with various lists
- ✚ HTML document with links to different parts of the same documents and to separate documents

### MySQL

- ✚ Table creation
- ✚ Data insertion and deletion
- ✚ Data retrieval
- ✚ Alteration of tables

## CSC4C04 – Visual Programming

Semester: 4

Course Number: 4

Contact Hours: 3T

Number of Credits: 3

Number of Contact Hours: 45T

**Course Evaluation:** Internal – 15 Marks + External – 65 Marks

**Aim of the Course:** To impart the students with the basic principles of visual programming.

### Objectives of the Course:

- ✚ To learn the basic concepts of .NET framework
- ✚ To learn the basics of visual programming using VB.NET

**Prerequisites:** Basic programming skills

### Course Outline

## **UNIT I (9T)**

Introduction to visual Programming -Concept of event driven programming, introduction to VB.Net, The .Net Framework and Common language runtime, Building VB.Net Application, VB IDE, forms, properties, events, VB language-console application and windows application, data type, declaring variable, scope of variable, operators and statements

## **UNIT II (9T)**

Control Statements- if-then, if -then- else, else-if ladder, select case, choose, loop statements- do loops, for, while-The with statement, converting between data types, Handling dates and times, Arrays - declaration and manipulation, Strings and String functions, procedures and functions

## **UNIT III (9T)**

Windows Applications-forms, adding controls to forms, handling events, MsgBox, Input Box, multiple forms, handling mouse and Keyboard events, object oriented programming- creating and using classes and objects, Handling Exceptions- on Error Goto

## **UNIT IV (9T)**

Common controls - textbox, Rich textbox, label, command Button, option button, checkbox, frame, list box, combo box, scrollbar, picture box, image box, timer, Data control, OLE, file controls-properties and methods

## **UNIT V (9T)**

Data Access with ADO.Net, accessing data with Server Explorer, Accessing Data with data Adaptors and Data sets, Creating a new data connection, creating and populating Data set, displaying data in Data Grid, selecting a data provider, Data accessing using Data adapter Control, Binding Data to Controls

## **References:**

1. *Visual Basic.NET Black Book*, Steven Holzner
2. *Visual Basic.NET for Developer*, Keith Franklin, Rebecca Riordan, SAMS.

3. *Learning Visual Basic.Net*, Jesse Liberty
4. *Visual Basic.Net Programming in Easy Steps*, Tim Anderson, Dreamtech Press

### **VB.NET Programming Lab List**

- + Simple VB.NET console applications
- + Simple VB.NET Windows applications
- + Programs involving different control structures
- + Programs involving various array operations
- + Programs involving various string functions
- + Programs involving forms and multiple forms
- + Programs involving mouse and keyboard events
- + Programs involving MsgBox, InputBox
- + Programs involving classes and objects
- + Programs involving exception handling
- + Programs involving common controls
- + Programs involving file controls
- + Programs involving OLE
- + Programs involving data access using ADO.NET

## **CSC4C05 – Programming Lab: C & Visual Programming**

**Semester:** 4

**Course Number:** 5

**Contact Hours:** 2L

**Number of Credits:** 2

**Number of Contact Hours:** 30L

**Course Evaluation:** Internal – 15 Marks + External – 65 Marks

**Aim of the Course:** To provide practical skill in Programming

## Objectives of the Course:

- ✚ To develop C Programming skills
- ✚ To develop VB.NET programming skills
- ✚ To develop Database handling skills
- ✚ To develop skills in using office software

**Prerequisites:** Background of the basic computing knowledge

## Course Outline

### LAB LIST

#### Word Processing

- ✚ Paragraph formatting
- ✚ Newspaper style Document
- ✚ Table creation
- ✚ Mail merge
- ✚ Page formatting & printing

#### Spreadsheet

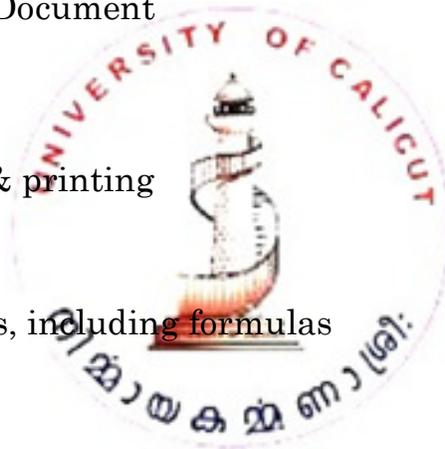
- ✚ Worksheet entries, including formulas
- ✚ Formatting cells
- ✚ Chart creation
- ✚ Functions

#### Presentation Software

- ✚ Creating presentation
- ✚ Animations
- ✚ Sound
- ✚ Inserting picture

#### C Programming

- ✚ Programs involving different data types
- ✚ Programs involving different arrays
- ✚ Programs involving different control structures and nested control



structures

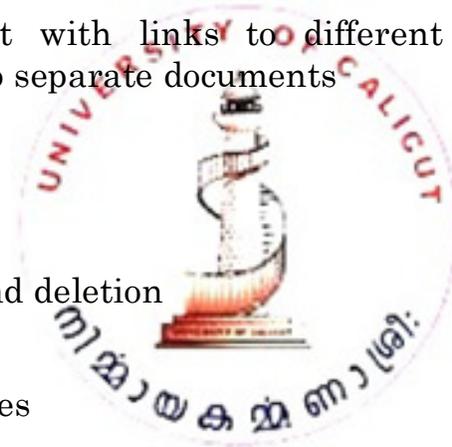
- # Programs involving functions
- # Programs involving recursion
- # Programs involving different storage classes
- # Programs involving strings
- # Programs involving structure and union
- # Programs involving files

## HTML

- # Simple HTML document creation
- # HTML document with tables
- # HTML document with various lists
- # HTML document with links to different parts of the same documents and to separate documents

## MySQL

- # Table creation
- # Data insertion and deletion
- # Data retrieval
- # Alteration of tables



## VB.NET Programming

- # Simple VB.NET console applications
- # Simple VB.NET Windows applications
- # Programs involving different control structures
- # Programs involving various array operations
- # Programs involving various string functions
- # Programs involving forms and multiple forms
- # Programs involving mouse and keyboard events
- # Programs involving MsgBox, InputBox
- # Programs involving classes and objects

- ✚ Programs involving exception handling
- ✚ Programs involving common controls
- ✚ Programs involving OLE
- ✚ Programs involving data access using ADO.NET

All lab works should be neatly recorded in a Laboratory Record Book in written form. However Program results can be pasted in the left hand side of the fare record. The laboratory record should have a minimum of:

- ✚ 20 lab exercises from C Programming
- ✚ 15 lab exercises from VB.NET Programming
- ✚ 3 exercises each from Word processing, Spreadsheet, Presentation Software, Web page design & Database

All students should maintain a rough record (observation note book) too, in which they write all the works to be carried out in the lab prior to his/her entry into the lab. He/She may also note down the i/p and o/p for program verification in the rough record.

