KANNUR UNIVERSITY
(Abstract)

B.Sc Computer Science-Scheme & Syllabus of Core and Complementary Courses under Choice Based Credit Semester System for Under Graduate Programmes-implemented with effect from 2009 admission-Orders Issued.

ACADEMIC BRANCH

No.Acad/C2/2389/2009 (1)   Dated, K.U.Campus. P.O,10-07-2009.

Read: 1.Minutes of the meeting of the Board of Studies in Computer Science (UG)
held on 26-05-2009.
2. Minutes of the meeting of the Faculty of Science held on 16-06-2009.
3. U.O No.Acad/C2/3838/2008 (i) dated 07-07-2009
4. Letter dated 02-07-2009 from the Chairman, BOS in Computer Science (UG).

ORDER

1.The Board of Studies in Computer Science(UG),vide paper read(1) above has prepared,finalized and recommended the Scheme and Syllabus of B.Sc Computer Science Core,Complementary and Open Courses under Choice Based Credit Semester System for implementation from 2009 admission.

2. The recommendations of the Board in restructuring the syllabus is considered by the Faculty of Science vide paper read (2) and recommended for the approval of the Academic Council.

3. The Regulations for Choice based Credit Semester System is implemented in this University vide paper read (3).

4. The Chairman, BOS in Computer Science (UG) vide paper read (4) above forwarded the restructured scheme and syllabus of B.Sc Computer Science Core and Complementary Courses under Choice Based Credit Semester System, prepared by the Board of Studies in Computer Science(UG) for implementation with effect from 2009 admission.

5. The Vice Chancellor, after examining the matter in detail, and in exercise of the powers of the Academic Council as per section 11(1) of Kannur University Act 1996 and all other enabling provisions read together with, has accorded sanction to implement the scheme and syllabus of B.Sc Computer Science Core and Complementary Courses

restructured in line with Choice Based Credit Semester System,with effect from 2009 admission, subject to ratification by the Academic Council.

6.The restructured scheme and syllabus of B.Sc Computer Science Core and Complementary Courses under Choice Based Credit Semester System, implemented with effect from 2009 admission is appended.

7. The Scheme and Syllabus of Complementary Courses offered for this Programme will be available along with the syllabus of Core Courses of the Complementary subject.

8.The affiliated Colleges are not permitted to offer Complementary Courses in violation to the provisional/permanent affiliation granted by the University. Changes in Complementary Courses are permitted with prior sanction /revision in the affiliation order already issued in this regard.

9. If there is any inconsistency between the Regulations for CCSS and its application to the Scheme & Syllabus prepared, the former shall prevail.

10. Orders are issued accordingly.

Sd/-
REGISTRAR

To:

1. The Principals of Colleges offering B.Sc Computer Science Programme.
2. The Examination Branch (through PA to CE)

Copy To:

1. The Chairman, BOS Computer Science (UG)
2. PS to VC/PA to PVC/PA to Regr
3. DR/AR I Academic
4.The Central Library
5. SF/DF/FC.

Forwarded/By Order

SECTION OFFICER

KANNUR UNIVERSITY

Course Structure
and
Syllabus

FOR

UNDERGRADUATE PROGRAMME

IN

COMPUTER SCIENCE

UNDER

CHOICE BASED CREDIT SEMESTER SYSTEM

w.e.f 2009 ADMISSION

Course Structure

B.Sc Computer Science

Semester – 1

Sl. No	Title of the Course	Hours/week		Credits
		Theory	Practical	
1	Common course English I	5		4
2	Common course English II	4		3
3	Common course Additional Language I	5		4
4	Informatics for Computer Science	1	2	4
4	Complementary I (Mathematics)	4		3
5	Complementary II	4		3

Semester – 2

Sl. No	Title of the Course	Hours/week		Credits
		Theory	Practical	
1	Common course English III	5		4
2	Common course English IV	4		3
3	Common course Additional Language II	5		4
4	Complementary I (Mathematics)	4		3
5	Complementary II	4		3
6	Core Course 1	1		2
7	Core Course 2		2	2

Semester 3

Sl No	Title of the Course	Hours/week		Credits
		Theory	Practical	
1	Common Course - Entrepreneurship	4		4
2	Common Course -Methodology of Computer Science	4		4
3	Complementary I (Mathematics)	5		3
4	Complementary II	5		3
5	Core Course 3	2		2
6	Core Course 4		5	3

Semester – 4

Sl No	Title of the Course	Hours/week		Credits
		Theory	Practical	
1	Common course - Numerical skills	4		4
2	Complementary I (Mathematics)	5		3
3	Complementary II	5		3

4	Core Course 5	3		3
4	Core Course 6	3		3
6	Core Course 7		5	3

Semester –5

Sl No	Title of the Course	Hours/week		Credits
		Theory	Practical	
1	Open Course I	2		2
2	Core Course 8	3		3
3	Core Course 9	3		3
4	Core Course 10	4		4
5	Core Course 11 (Elective I)	5		4
6	Core Course 12		4	2
7	Core Course 13		4	2

Semester –6

Sl. No.	course	Hours/week		Credits
		Theory	Practical	
1	Open course II	2		2
2	Core Course 14	3		3
3	Core Course 15	3		3
4	Core Course16	3		3
5	Core Course 17 (Elective II)	4		4
6	Core Course 18		8	0
7	Core Course 19		2	5

The distribution of Credits/Hours for Complementary Theory/Practical shall be decided by the Board of Studies concerned.

Scheme Core(Computer Science)

No	Semester	Course Code	Title of the course	Hours / Week	Credits
1	I	1A13CSC	Informatics for Computer Science	3	4
2	II	2B01CSC	Advanced Programming in C	2	2
3	II	2B02CSC	Lab - II (Advanced Programming in C)	2	2
4	III	3B03CSC	Programming in C++	2	2
5	III	3B04CSC	Lab - III (Programming in C++ & Data structures)	5	3
6	III	3A11CSC	Entrepreneurship	4	4

7	III	3A14CSC	Methodology of Computer Science	4	4
8	IV	4B05CSC	Visual Programming	3	3
9	IV	4B06CSC	Database Management Systems	3	3
10	IV	4B07CSC	Lab - IV (Visual programming &DBMS)	5	3
11	IV	4A12CSC	Numerical skills	4	4
12	V	5B08CSC	Software Engineering	3	3
13	V	5B09CSC	Web Technology	3	3
14	V	5B10CSC	Java Programming	4	4
15	V	5B11CSC	Elective I	5	4
16	V	5B12CSC	Lab -V (Java programming)	4	2
17	V	5B13CSC	Lab -VI (Web Technology)	4	2
18	VI	6B14CSC	Data Communication and Computer Network	3	3
19	VI	6B15CSC	Computer Organization	3	3
20	VI	6B16CSC	Operating Systems	3	3
21	VI	6B17CSC	Elective II	4	4
22	VI	6B18CSC	Lab-VII (Software Development)	8	0
23	VI	6B19CSC	Project	2	5

The syllabus for 3A11CSC and 4A12CSC shall be framed and communicated later.

Electives

Note : Course 5B11CSC shall be selected from Section A and Course 6B17CSC from section B of the following list of elective courses.

No	Semester	Course Code	Title of the course	Hours/ Week	Credits
Section A					
1	V	5B11CSC E01	Algorithm Analysis & Design	5	4
2	V	5B11CSC E02	Network Programming	5	4
3	V	5B11CSC -E03	Soft computing.	5	4
4	V	5B11CSC- E04	Numerical Methods	5	4
5	V	5B11CSC -E05	Computer Graphics	5	4

6	V	5B11CSC- E06	Digital Systems and Microprocessors	5	4
Section B					
7	VI	6B17CSC E07	Information Security	4	4
8	VI	6B17CSC E08	Mobile Communication	4	4
9	VI	6B17CSC E09	Data Mining.	4	4
10	VI	6B17CSC E10	C# and .NET frame work	4	4
11	VI	6B17CSC E11	Digital Image Processing	4	4
12	VI	6B17CSC E12	Data Compression	4	4
13	VI	6B17CSC E13	Linux Administration	4	4

Scheme Complementary (Computer Science)

No	Semester	Course Code	Title of the course	Hours / Week	Credits
1	I	1C01CSC	Introduction to IT and C programming	2	2
2	I	1C02CSC	Lab – I (C Programming)	2	1
3	II	2C03CSC	Programming in C++	2	2
4	II	2C04CSC	Lab – II (C++ Programming)	2	1
5	III	3C05CSC	Data Base Management System	3	2
6	III	3C06CSC	Lab III (DBMS Lab)	2	1
7	IV	4C07CSC	Visual Basic	3	2
8	IV	4C08CSC	Lab IV (Visual Basic)	2	1

Scheme- Open Courses

No	Semester	Course Code	Title of the course	Hours/ Week	Credits
1	V	5D01CSC	C programming -I	2	2
2	V	5D02CSC	Introduction to Information Technology	2	2
3	V	5D03CSC	Computer Application Packages - I	2	2
4	V	5D04CSC	Programming in C++ - I	2	2
5	V	5D05CSC	Programming in JAVA - I	2	2
6	V	5D06CSC	Numerical Methods - I	2	2
7	V	5D07CSC	Data Base Management System - I	2	2
8	V	5D08CSC	Web Technology - I	2	2
9	VI	6D01CSC	C programming -II	2	2
10	VI	6D02CSC	Programming in C++ - II	2	2
11	VI	6D03CSC	Computer Application Packages - II	2	2
12	VI	6D04CSC	Programming in Java II	2	2
13	VI	6D05CSC	Data Base Management System - II	2	2
14	VI	6D06CSC	Numerical Methods - II	2	2
15	VI	6D07CSC	Web Technology - II	2	2

Evaluation

Continuous Evaluation

Continuous assessment carries a weightage of 25%. There shall be no separate minimum for CE.

The components of continuous evaluation are given below :

a. Theory

Component			Weight
a	Attendance	Below 75% - F; 75 – 79 : D; 80 – 89 : C 90 – 94 : B; 95 – 100 : A	1
b	Assignment	One or more assignments shall be given. The number of assignment shall be proportional to the effort required to complete the assignments. Evaluation criteria shall be decided by the faculty concerned and must be made available to the students at the beginning of the semester. Structure, content, presentation, timely submission etc shall be considered for awarding grade.	1
c	Seminar / Viva	Students may be asked to take a seminar on a topic relevant to the course (Not from the prescribed syllabus). Evaluation criteria shall be decided by the faculty concerned and must be made available to the students at the beginning of the semester. Seminar report, presentation skill, preparation etc shall be considered for awarding grades. **OR** The faculty may conduct (himself/ herself or as a team) one or more course viva based on the syllabus of the course.	1
d	Tests	A minimum of two tests shall be conducted. Test papers shall be graded by the same procedure adopted for ESE. If more than two tests are conducted, best two grades shall be considered for calculation of CE grade	2

Consolidation of Grades for CE (Theory) - Sample

Component	**Weight (W)**	**Grade Awarded**	**Grade points (G)**	**Weighted Grade points (WxG)**
Attendance	1	A	4	4
Assignment	1	B	3	3
Seminar / Viva	1	B	3	3
Test paper	2	C	2	4
Total	5			14
CE Grade	Total weighted grade points / Total weights = 14/5 = 2.8 = B			

b. Practical

Component			Weight
a	Attendance	Below 75% - F; 75 – 79 : D; 80 – 89 : C 90 – 94 : B; 95 – 100 : A	1
b	Practical test / Lab skill	Performance of the students in the practical sessions may be evaluated regularly. *Evaluation criteria shall be decided by the faculty in charge and should be made available to the students at the beginning of the semester.* Grade awarded shall be based on preparation for practical session, maintenance of rough record, diligence, timely completion of exercises etc.	2
c	Practical record	Grade shall be awarded based on recording of required number of lab assignments, format, content, Presentation and neatness of the record, timely submission etc. Evaluation criteria should be decided by the *faculty in charge* and should be made available to the students at the beginning of the semester.	1
d	viva	Viva may be conducted on regular basis or at the end of the semester. The details should be made available at the beginning of the semester by the faculty in charge.	1

Consolidation of Grades for CE (Practical) – Sample

Component	Weight(W)	Grade Awarded	Grade points (G)	Weighted Grade points (WxG)
Attendance	1	A	4	4
Lab skill / practical test	2	B	3	6
Record	1	B	3	3
Viva	1	C	2	2
Total	5			15
CE Grade	Total weighted grade points / Total weights = 15/5 = 3 = B			

End Semester Evaluation

i. Examination for both theory and practical courses shall be conducted at the end of the respective semester.

ii. Duration of examination for both theory and practical, unless otherwise specified in the syllabus of the course, shall be **three** hours.

iii. End – semester evaluation in practical courses shall be conducted by two examines – one internal and one external.

iv. End Semester Examination question shall consist of the following types and number of questions :

Sr No	Type of questions	weight	Number of Questions to be answered / total number of questions	Max. weighted grade point
1	A bunch of four objective type questions	1	2 bunch x 4 = 8	8
2	Short answer	1	5 / 8	20
3	Short essay / programs	2	5/8	40
4	Essay type	4	1/2	16

1A13CSC Informatics for B Sc Computer Science

Contact Hours per Week : 1 Theory + 2 Practical
No. of Credits :4

Note : The paper consists of theory as well as practical components. End Semester Assessment will be based only on the theory component. Internal assessment shall include practical assignments.

Aim of the Course

To update and expand basic informatics skills and attitudes relevant to the emerging knowledge society and also to equip the students to effectively utilize the digital knowledge resources for their chosen courses of study.

Objectives of the Course

• To review the basic concepts & functional knowledge in the field of informatics.
• To review functional knowledge in a standard office package and popular utilities
• To create awareness about nature of the emerging digital knowledge society
• To create awareness about social issues and concerns in the use of digital technology
• To impart skills to enable students to use digital knowledge resources in learning.

A. Theory

<u>Module I</u> : (Ref Essential reading 1)

Computer basics ; Evolution, generation and classification of computers.

Computer Organization and Architecture : CPU; Communication among various units; Instruction format; Instruction cycle; Instruction set; Data representation; coding schemes.

Computer memory and Storage : Memory hierarchy; RAM; ROM; secondary storages- magnetic, optical and magneto-optical storage devices. Mass storage devices. Input output devices -Types of I/O devices.

<u>Module II</u> : (Ref Essential reading 2)

Computer Programming : Algorithm; Flow chart.

Program : structure, top-down design, source code, object code, executable file, file extensions. Importance of C; Basic structure of C, Programming style, executing a c program. Character set, C tokens, Keywords, identifiers, Constants, data types, declaration of variables, arithmetic operators , logical operators, Relational operators, Assignment operators, Increment and decrement operators, conditional operators, Bitwise operators. Precedence and order of evaluation. type conversion in expression.

common programming errors, program testing and debugging.

Module III : (Ref Essential reading 2)

Managing Input output operation: reading a character, writing a character, formatted input output. Branching statements-if, if..else, nested if...else, else...if ladder, switch statement, go to statement. Looping statements- while, do...while, for loop. Break and continue statements.

Module IV - KNOWLEDGE SKILLS FOR HIGHER EDUCATION

Data, information and knowledge, knowledge management- Internet access methods – Dial-up,DSL, Cable, ISDN, Wi-Fi - Internet as a knowledge repository, academic search techniques,Creating cyber presence, **(case study of academic websites), open access initiatives, openaccess publishing models. Basic concepts of IPR, copyrights and patents, plagiarism,introduction to use of IT in teaching and learning, **(case study of educational software), **(academic services-INFLIBNET, NICNET, BRNET)

Module V - SOCIAL INFORMATICS

IT & Society- issues and concerns- digital divide, IT & development, the free software movement , IT industry: new opportunities and new threats, software piracy, cyber ethics, cybercrime, cyber threats, cyber security, privacy issues, cyber laws, cyber addictions, informationoverload, health issues- guide lines for proper usage of computers, internet and mobile phones. impact of IT on language & culture-localization issues- Unicode-IT and regional languages

Note :

i. Self study / seminars / group discussion/ demonstrations shall be employed for modules IV and V.

ii. ** .To be excluded from End semester evaluation.

Essential Reading

1. Introduction to information Technology, ITL Education solutions, Pearson Education
2. ANSI C, E. Balagurusamy, 3rd edition McGraw-Hill Publication
3. V. Rajaraman, Introduction to Information Technology, Prentice Hall
4. Technology in Action, Pearson
5. Alexis Leon & Mathews Leon, Computers Today, Leon Vikas,
6. Peter Norton, Introduction to Computers,6e,(Indian Adapted Edition)
7.

Additional References

7. Greg Perry, SAMS Teach Yourself Open Office.org, SAMS,
8. Alexis & Mathews Leon, Fundamentals of Information Technology, Leon Vikas
9. George Beekman, Eugene Rathswohl, Computer Confluence, Pearson Education,
10. Barbara Wilson, Information Technology: The Basics, Thomson Learning

11. John Ray, 10 Minute Guide to Linux, PHI, ISBN 81-203-1549-9
12. Ramesh Bangia, Learning Computer Fundamentals, Khanna Book Publishers
13. Computer Basics and c Programming, V. Rajaraman, PHI, 2008
14. Let us C, Yeshvanth Kanethkar, 3rd Edn, BPB,
15. Programming with C in Linux, NIIT, PHI.

B. Practical

List of Programs

Students have to do and record a minimum of 15 programs. They have to be familiar with both windows and Linux platforms.

1. Develop a program that reads a floating point number and then displays the right most digit of the integral part of the number.
2. Develop a program to read an integer and print sum of digits of the number.
3. Develop a program to read a lower case English alphabet and print equivalent upper case alphabet.
4. The straight line method of computing the yearly depreciation of the value of an item is given by

   $$Depreciation = \frac{purchase\ price - Salvage\ value}{Year\ of\ Service}$$

   Develop a program to determine the salvage value of an item when the purchase price, year of service and annual depreciation are given.
5. Develop an interactive program to demonstrate the process of multiplication. The program should ask the user to enter two two-digit integers and print the product of integers as shown below:

```
                    45
               X    37
              ___________
   7 x 45 is        315
   3 x 45 is        135
              ___________
   Add them        1665
              ___________
```

6. Develop a program to find the number of and sum of all integers grater than 100 and less than 200 that are divisible by 7.
7. A set o two linear equations with two unknowns x1 and x2 is given below:

   $$a\ x_1 + b\ x_2 = m$$
   $$c\ x_1 + d\ x_2 = n$$

   The set has a unique solution

   $$x1 = \frac{md-bn}{ad-cb} \qquad x2 = \frac{na-mc}{ad-cb}$$

   Provided the denominator ad – cb is not equal to zero.
   Develop a program that will read the values of the constants a, b, c, m and n and compute the values of x1 and x2. An appropriate message should be printed if ad – cb = 0.

8. Admission to a professional course is subject to the following conditions:
    a. Marks in mathematics >=60.
    b. Marks in Physics >=50.
    c. Marks in Chemistry >=40.
    d. Total in all three subjects >=200 Or
       Total in Mathematics and Physics >=150.

Given the marks in the three subjects, develop a program to print whether an applicant is eligible or not.

9. A cloth showroom has announced the following seasonal discounts on purchase of items:

Purchase amount	Discount	
	Mill cloth	Handloom
Items		
0 – 100	--	5%
101 – 200	5%	7.5%
201 – 300	7.5%	10%
Above 300	10%	15%

Develop a program using switch and if statements to compute the net amount to be paid by a customer.

10. Develop a program that will read the value of x and evaluate the following function :

$$y = \begin{cases} 1\, for\, x < 0 \\ 0\, for\, x = 0 \\ -1\, for\, x < 0 \end{cases}$$ using nested if, else if and conditional operator.

11. Develop a program using do—while loop to print the first m Fibonacci numbers.
12. Develop a program to read n numbers and print number of odd numbers in the list.
13. Develop a program to print the multiplication table of a given number.
14. Develop a program to print the following output using for loop :

```
*******
******
*****
****
***
**
*
```

15. Develop a program to read the age of 100 persons and count the number of persons in the age group 50 to 60. Use for and continue statements.
16. . Develop a program to read a positive integer and print its binary equivalent.

17.. Develop a program to compute Euler’s number e using the following formulae :
E = 1+1/1! +1/2!+1/3! + ….+1/n!

18. Develop a program to evaluate the following function to 0.0001% accuracy.
$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots.$

19. Develop a program to sum first n terms of the series S = 1 + 3 + 4+7+11+…

20. Develop a program to read two integers and an operator (+, -, * and /). The program should perform the operation specified and print the result. Use switch statement.

MODEL QUESTION PAPER

1A13CSC INFORMATICS FOR COMPUTER SCIENCE

Time: 3 Hours **Maximum Weighted Grade Point=84**

SECTION A

Answer all Questions. Weightage for a bunch of four questions is 1.
Maximum Weighted Grade Point 1(W) x 2 (bunch) x 4(Max GP)= 8

1. A register that keep track of next instruction to be executed is called a ….
   a. Program counter b. Instruction register c. accumulator d. Data register
2. RAID stands for ………………
3. Which of the following is incorrect declaration ?
   a. auto int count; b. register char ch; c. extern long total; d. static name;
4. What is the value of the following statement ? -14%-3
   a. -2 b. +2 c. 1 d. 0
5. For using character functions, we must include the header file ………. In the program.
6. A for loop with no test conditions is known as -- loop.
7. DSL stands for ………….
8. Madhuri is a software for ……………

SECTION B

Answer any 5 Questions. Weightage 1 each
Maximum Weighted Grade Point 1(W) x 5(Qn) x 4(Max GP)=20

9. Give the hierarchy of compute memory.
10. Define instruction cycle.
11. List the logical operators in C.
12. Define algorithm
13. What is a compiler?
14. Write if statement required to find the largest of three numbers : I, j and k.
15. Define data and information.
16. What is Unicode?

SECTION C

Answer any 5 Questions. Weightage 2 each
Maximum Weighted Grade Point 2(W) x 5(Qn) x 4(Max GP)=40

17. Write note on optical storage devices.
18. Discuss hierarchy of operators and expression evaluation in c.
19. Draw a flow chart to find the largest of three numbers.
20. Write c program to read n positive integers and print number of odd and even numbers.
21. Discuss IPR.
22. Discuss use of ITin education.
23. Write notes on cyber ethics.

24. Discuss about proper usage of internet.

SECTION D

Answer any one Question Weightage 4

Maximum Weighted Grade Point 4(W) x 1(Qn) x 4(Max GP)=16

25. a. Discuss internet as a knowledge repository.
   a. Write notes on cyber crime.

26. Write notes on the following :
   a. Switch statement
   b. While and do while constructs
   c. Formatted IO in c.

2B01CSC Advanced programming in C

Contact Hours per Week : 2 Theory
Credit : 2

Objectives :

1. Introduce advanced programming concepts using C.
2. To develop c programs using advanced constructs.
3. To design algorithm for solving a programming problems.
4. Develop skill in programming.

Module I Arrays: One dimensional arrays, two dimensional arrays, Initializing array elements, Multidimensional arrays.

Strings : declaration and initializing , reading and writing. Arithmetic operations on character. String handling functions.

Module II Functions: Library and user defined, defining a function, calling a function. Parameter passing techniques, Scope and life time of variables in function, recursive functions, arrays and functions.

Module III Structure and union: definition, giving values to members, initialization. Array of structures, array with in structure, structure with in structure, union.

Module IV Pointers: accessing the address of a variable, declaration and initializing pointers, accessing a variable through its pointers, pointer arithmetic, pointers and arrays (pointer to array and array of pointers) , pointers and character string , pointer and functions. Dynamic memory allocation: malloc(), calloc(), free(),realloc().

Module V File Management: Text and binary files, Defining and opening a file, closing a file, input and output operations on file, error handling, random access file. Bitwise operations; Preprocessor directives, #include, #define, Macros with arguments; Conditional compilation. Header file concept. Multiple file programming. Command line arguments.

Text Book :

1. ANSI C, E. Balagurusamy, 3rd edition McGraw-Hill Publication

Reference books:

1. Programming with ANSI and Turbo C, Ashok N. Kamthane, 1st edn, Pearson Education.
2. Let us C, Yeshvanth Kanethkar, 3rd Edn, BPB,
3. Programming with C in Linux, NIIT, PHI.
4. C by Example, Noel Kalicharan, Cambridge University press.

Web Resources:

1. www.cprogramming.com
2. www.programmersheaven.com

Question Distribution

Module	Number of questions			
	Section A	Section B	Section C	Section D
I	1	2	2	2
II	2	1	2	
III	1	2	1	
IV	2	1	1	
V	2	2	2	

MODEL QUESTION PAPER

2B01CSC Advanced programming in C

Time: 3 Hours **Maximum Weighted Grade Point=84**

SECTION A

Answer all Questions. Weightage for a bunch of four questions is 1.
Maximum Weighted Grade Point 1(W) x 2 (bunch) x 4(Max GP)= 8

1. Maximum number of elements in the array declaration int a[5][8] is
   a. 28 b. 32 c. 35 d. 40
2. For using character functions, we must include the header file ………. In the program.
3. A function that call itself is known as a ……….. function.
4. A …….. is a collection of data items under one name in which the items share the same storage.
   a. Structure b. Array c. Union. D. Strings
5. A pointer holds :
   a. Address of a variable b. Value of a variable
   b. Both (a) and (b) d. None of the above.
6. The function malloc() is used for …………
7. The mode … is used for opening a file for updating.
8. When the main function is called, it is called with the arguments :
   a. argc b. argv c. Both a and b d. None of these

SECTION B

Answer any 5 Questions. Weightage 1 each

Maximum Weighted Grade Point 1(W) x 5(Qn) x 4(Max GP)=20

9. Write necessary array declaration statements: list of 100 integers, a matrix of size 10x20 and a list of 100 names.
10. What is a string?
11. What is a static variable?
12. Give suitable examples for array of structures and array within structure.
13. Differentiate between structure and union.
14. Explain the following : int *k = malloc (sizeof int);
15. Differentiate between text and binary files.
16. What is a macro?

SECTION C

Answer any 5 Questions. Weightage 2 each

Maximum Weighted Grade Point 2(W) x 5(Qn) x 4(Max GP)=40

17. Write a c program to read a list of n names, sort them in alphabetic order and print the modified list.
18. Write a program to read n integers and print their mean and standard deviation.
19. With suitable example(s), explain parameter passing techniques in c functions.
20. Write a function to find the sum of diagonal elements of an nxn matrix.
21. Declare a structure with fields : name, age and amount (real number). Write a program to read a record and print it.
22. Explain how a one dimensional array can be accessed with pointers. With suitable example discuss pointer arithmetic.
23. Write a program to create a text file. Each record of the file should contain a name and a number.
24. With suitable example program, explain command line arguments.

SECTION D

Answer any one Question Weightage 4

Maximum Weighted Grade Point 4(W) x 1(Qn) x 4(Max GP)=16

25. Write a complete program to read two matrices and print their product. Your program should be general and well structured. (Use function for various sub tasksl)
26. With suitable examples explain the following:
    a. Dynamic memory allocation.
    b. Preprocessor directives
    c. Conditional compilation.
    d. Pointer to array and array of pointers.

2B02CSC Lab II (Advanced Programming in C)

Contact Hours per Week : 2 Practical
Credit : 2

<u>List of Programs</u>

Students have to do and record a minimum of 15 programs. They have to be familiar with both windows and Linux platforms.

1. Develop a program to read n integers and print their mean and standard deviation.
2. Develop a program to sort a list of n positive integers in ascending order.
3. Develop a program to search a list of integers for a key k. (Sequential search).
4. Given two one dimensional sorted (ascending) arrays A and B. Develop a program to merge them into a single sorted array C that contains every item from arrays A and B, in ascending order.
5. Develop a program for matrix multiplication.
6. Write a program for fitting a straight line through a set of points (x_i, y_i), i=1,2,…n. The straight line equation is y = mx +c. the values of m and c are given by:

$$m = \frac{n\sum(x_i y_i) - (\sum x_i)(\sum y_i)}{n(\sum x_i^2) - (\sum x_i)^2} \qquad c = \frac{1}{n}(\sum y_i - m\sum x_i)$$

7. Design and code an interactive modular program that will use functions to read a matrix of m by n size, compute column averages and row averages, and then print the entire matrix with averages shown in respective rows and columns.
8. Develop a program to read a string and determine whether it is a palindrome.
9. Develop a program to read a text and count occurrences of a particular word.
10. Develop a program to replace a particular word by another word in a given string.
11. Develop a program to read a set of n names and sort them in alphabetic order.
12. Develop a function prime that returns 1 if its argument is a prime number and returns a zero otherwise. Develop a main program to read n integers into an array and print all prime numbers in the array.
13. Develop a recursive function to find factorial of a number.
14. Develop a function that will scan a character string passed as an argument and convert all lower case characters into their uppercase equivalent.
15. Define a structure data type called time-struct containing three fields : integer hour, integer minute and integer second. Develop a program that would assign values to the individual members and display the time in the following form: 16:30:21
16. Define a structure named census with the following thre members :

- A character array city[] to store names.
- A long integer to store population of the city.
- A float member to store the literacy level.

Develop a program to do the following:

- To read details for 5 cities.
- To sort the list of cities in alphabetic order.
- To sort the list based on population.
- To display the sorted lists.

17. Develop a function (using pointers) that reverses the elements of a given array.
18. Develop a program that creates a text file. Each record contains three fields, roll number, name and age. Develop a program to read the created file and print its contents. Format your output suitably.
19. Develop a program to copy the contents of one file into another.
20. Two files DATA1 and DATA2 contain sorted lists of integers. Develop a program to produce a third file DATA which holds a single sorted, merged list of these two lists. Use command line arguments to specify the file names.

3B03CSC Programming in C++

Contact Hours per Week : 2 Theory
Credit : 2

Aim : To introduce Object oriented concepts and to impart skill in object oriented programming using C++.

Objectives: 1. Introduce concepts such as classes and objects.
2. Define and use classes and objects using C++ language.
3. Introduce OOPs concepts such as inheritance and polymorphism and their implementation using C++.
4. Skill in developing OO Program for a given problem.

Module I : Principles of object oriented programming; OOP paradigm; Basic concepts of OOP; Benefits; applications.
Introduction to C++, Structure of C++ program; how to create and execute a C++ program under Windows and Linux.
Tokens, Keywords, identifiers and constants; Basic data types; user defined data types; Derived data types; symbolic constants; type compatibility; declaration and dynamic initialization of variables; reference variables.
Operators; Scope resolution; memory dereferencing and memory management operators; manipulators; type cast operators; Expressions and their types; Special assignment expressions; implicit conversions; operator overloading; operator precedence; Control structures.

Module II : Functions : main; prototyping; call by reference; inline function; default and const arguments; function overloading; friend and virtual functions; Math library functions.
Structures; Specifying a class; Defining member functions; making an outside function inline; nesting of member functions; private member functions; arrays within a class; memory allocation for objects; static data members; static member functions; arrays of objects; objects as function arguments; friendly functions; returning objects; const member functions; pointer to members; Local classes.

Module III :Constructors and destructors : - Constructors; Parameterized constructors; Multiple constructors; multiple constructors in a class; constructors with default arguments; dynamic initialization of objects; copy constructor; Dynamic constructors; const objects; Destructors.
Operator overloading – definition; overloading unary operators; overloading binary operators; overloading binary operators using friends; manipulation of strings using operators; rules for overloading operators. Type conversions.

Module IV :Inheritance – defining derived classes; single inheritance; making a private member inheritance; multilevel inheritance; multiple inheritance; hierarchical inheritance; hybrid inheritance; virtual base classes; abstract classes; constructors in derived classes; Nesting of classes.Pointers; Pointers to objects; Pointers to derived classes; virtual functions; pure virtual functions.

Module V :C++ streams; stream classes; unformatted I/O operations; Formatted console I/O operations; Managing output with manipulators.
Files – classes for file stream operations; Opening and closing a file; file modes; file pointers and their manipulations; Sequential input and output operation.

Text book :

1.Object Oriented Programming with C++; E. Balagurusamy; 3rd Edn; TMH 2006

Reference Books: .

1. Programming in C++, M.T. Somashekara, Prentice Hall of India, New Delhi

2.Object Oriented Programming with ANSI & Turbo C++, Ashok N. Kamthane, Pearson Education
3.Let us C++, Yeshwanth Kanethkar, BPB

3B04CSC Lab III (Programming in C++ and Data structures)

Contact Hours per Week : 5 Practical
Credit : 3

The lab consist of two sections, Section A : Programming in C++ and B : Data structures. Equal weightage will be given for both sections. For internal assessment, each part may be evaluated independently and final CA grade shall be obtained by combining them. End semester examination question shall carry questions from both sections.

3A14CSC Methodology of Computer Science

Contact Hours per Week : 4 Theory
Credit : 4

Aim : Familiarize the students with the methodology of computer science.

Objectives :

1. To familiarize students with concept of data structures and its relevance in computer science.
2. To introduce the concept of analysis of algorithms and ability ro compare algorithms based on time and space complexity.
3. To familiarize with selected linear and nonlinear data structures.
4. To enhance skill in programming.
5. To inculcate systematic approach to programming.
6. Develop ability to select appropriate data structure for a given problem.
7. Develop ability to design new data structures.

Syllabus

Module I Data structures: Definition and Classification.

Analysis of Algorithms : Apriori Analysis; Asymptotic notation; Time complexity using O notation; Average, Best and Worst complexities.

Arrays :- Operations; Number of elements; Array representation in memory.

Polynomial- Representation with arrays; Polynomial addition.

Sparse Polynomial:- representation.

Sparse matrix: Efficient representation with arrays; Addition of sparse matrices.

Recursive algorithms: examples – factorial and Tower of Hanoi problem.

Module II Search : Linear and Binary search; Time complexity; comparison.

Sort : Insertion, bubble, selection, quick and merge sort; Comparison of Sort algorithms.

Module III Stack: Operations on stack; array representation.

Application of stack- i. Postfix expression evaluation. ii. Conversion of infix to postfix expression.

Queue : Operation on queue. Array Implementation; Limitations; Circular queue; Dequeue and priority queue. Application of queue: Job scheduling.

Module IV Linked list – Comparison with arrays; representation of linked list in memory.

Singly linked list- structure and implementation; Operations – traversing/printing; Add new node; Delete node; Reverse a list; Search and merge two singly linked lists.

Stack with singly linked list.

Circular linked list – advantage. Queue as Circular linked list.

Head nodes in Linked list – Singly linked list with head node – Add / delete nodes; Traversal / print.

Doubly linked list – structure; Operations – Add/delete nodes ; Print/traverse. Advantages.

Module V 10 Hrs

Tree and Binary tree: Basic terminologies and properties; Linked representation of Binary tree; Complete and full binary trees; Binary tree representation with array. Tree traversal : Recursive inorder, preorder and postorder traversals. Binary search tree - Definition and operations (Create a BST, Search, Time complexity of search). Application of binary tree: Huffman algorithm.

Text Book :

1. Data Structures and Algorithms: Concepts, Techniques and Applications; GAV Pai, Mc Graw Hill, 2008

Reference Books :

2. Data Structures in C, Achuthsankar and Mahalekshmi, PHI, 2008
3. Fundamentals of Data structures in C++ , 2nd Edn, Horowitz Sahni, Anderson, Universities Press
4. Classic Data structures, Samanta, Second Edition, PHI

Web Resources :

1. *http://www.cs.umd.edu/~mount/451/Lects/451lects.pdf*
2. *http://www.brpreiss.com/books/opus6/html/page37.html*
3. *www.nist.gov/dads/HTML/*
4. *http://www.inversereality.org/tutorials/c++/linkedlists.html*
5. *http://www.cs.usask.ca/resources/tutorials/csconcepts*
6. http://oopweb.com/Algorithms/Files/Algorithms.html
7. http://www.cs.ubc.ca/spider/harrison/Java/sorting-demo.html

4B05CSC Visual Programming

Contact Hours per Week : 3
Credit : 3

Objectives :

i. To introduce Widows programming environment.
ii. To familiarize with Microsoft foundation Classes.
iii. Skill in developing programs with VC++.
iv. Introduce database connectivity using ODBC.

Module I The windows environment, dynamic linking, your first windows program, A character – mode model, windows equivalent , Header files ,The MessageBox() function, An introduction to Unicode, brief history, Windows and messages, A window of one's own, An Architectural overview ,Registering the window class, Creating and displaying the window, the message loop, processing the messages, WM_PAINT and WM_DESTROY messages, An Introduction to GDI: Device context, Getting Device context, basic drawing, Structure of GDI primitives, Drawing basic shapes.

Module II Overview of MFC programming: What is MFC, Advantages of using MFC, features, MFC fundamentals: Class hierarchy, member functions, global functions, AFXWIN.H, MFC application skeleton, creating frame windows, CWinApp, Processing messages: Responding to messages- MFC style, BEGIN_MESSAGE_MAP() macro, Responding to key press, MFC Device context classes, WM_PAINT, WM_DESTROY, DrawText() and TextOut() functions, simple graphics programs.

Module III Introducing menus: Using Resources, compiling .RC files, creating a menu, Responding to Menu commands, Keyboard: keyboard message handlers , virtual key code, samlple programs for handling keyevents, Mouse: Handlers, handling mouse events program. Toolbar: CToolbar class.

Module IV Dialog Based applications: creating a dialog based programs, modal and modaless dialog boxes, the CDialog class,DoDataExchange(), UpdateData().OnInItDialog(), DoModal(), OnOk(), OnCancel() functions.
Writing simple dialog based programs.

Module V

ODBC classes: ODBC,database drivers , data source name, connecting visual C++ programs to remote database, CDataBase class, Open(), Close() functions, CRecordSet class, establishing connection, MoveFirst(), MoveNext(), MovePrev(), MoveLast functions, adding, editing and deleting records. M_strSort and m_StrFilter variables. creating simple database editing programs.

Texts:

Module I—Programming windows by Charles PetZold
Module II ---VC++ 6 from the ground up by John Paul Muller
Module III ---VC++6 programming by Yaswant Kanitkar
Module IV -mastering VC++ by Micheal J.Young BpB publications
Module V -MFC programming with VC++6 by David white,Kennscribner,Eugene olafsen.

4B06CSC Database Management System

Contact Hours per Week: 3
Credit : 3

Objectives :

1. Introduce the basic concepts in DBMS.
2. Skill in designing database.
3. Familiarization of different DBMS models.
4. Skill in writing queries using MySQL.

Module I Introduction – purpose of Database systems. View of Data, data Models, transaction management, database structure, DBA, Data Base Users.

Module II E-R model, Basic concepts; design issues; Mapping Constraints; Keys; Primary, Foreign, candidate, E-R diagram; Weak entity set; Extended E-R features.
Normal forms – 1NF, 2NF, 3NF and BCNF; functional dependency, Normalization.

Module III SQL : database languages; DDL; create, alter, Drop, DML, Insert into, Select, update, Delete,. DCL commands, Data types in SQL; Creation of database and user.
Case study : MySQL.

Module IV Developing queries and subqueries; Join operations; Set operations; Integrity constraints, views, Triggers, functions and Sequences.
Case study : MySQL.

Module V Relational model – Structure of Relational database. Relational Algebra; Fundamental operations; Relational calculus; Tuple and domain calculus.

Text books:

1. Database system concepts; Silberschatz, Korth and Sudarsan, 5th Edn; McGraw Hill.
2. The database book : Principles and Practice Using MySQL; Gehani; University Press.

Reference:

1. Fundamentals of Database systems, E. Navathe, 4th edn, Pearson Education.

4B07CSC Lab IV (Visual programming and DBMS)

Contact Hours per Week: 5 Practical
Credit : 3

The lab consist of two sections, Section A : Visual programming and B : DBMS. Equal weightage will be given for both sections. For internal assessment, each part may be evaluated independently and final CA grade shall be obtained by combining them. End semester examination question shall carry questions from both sections.

5B08CSC Software Engineering

Contact Hours per Week: 3
Credit : 3

Aim : To introduce basics of software engineering.

Objectives :

1. Understand the basic processes in software Development life cycle.
2. Familiarize with different models and their significance.
3. Approach software development in a systematic way.
4. To familiarize students with requirement engineering and classical software design techniques.
5. To introduce objected oriented design concepts.
6. To familiarize with various software testing techniques and tools.

Module 1: Introduction to software engineering-Definition, program versus software, software process, software characteristics, brief introduction about product and process, software process and product matrices; Software life cycle models – Definition, waterfall model, increment process model, evolutionary process model, selection of the life cycle model.

Module 2: Software Requirement Analysis and Specification – Requirements engineering, types of requirements, feasibility studies, requirement elicitation, various steps of requirement analysis, requirement documentation, requirement validation.
** [An example which illustrate various stages in requirement analysis.]

Module 3: Software design – definition, various types, objectives and importance of design phase, modularity, strategy of design, function oriented design, IEEE recommended practice for software design descriptions.

Module 4: Objected Oriented Design – Analysis, design concept, design notations and specifications, design methodology.
**[case study based on Objected Oriented Design]

Module 5: Software Testing – What is testing, Why should we test, who should do testing? Test case and Test suit, verification and validation, alpha beta and acceptance testing, functional testing , techniques to design test cases , Boundary value analysis, equivalence class testing, decision table based testing , cause effect graphing techniques ; structural testing , path testing , cyclomatic complexity , Graph matrices , Data flow testing , mutation testing , levels of testing , unit testing , integration testing , system testing , validation testing , a brief introduction about debugging and various testing tools.
(10 hours)

**** Topics not to be included for end semester evaluation. Nevertheless they may be included for continuous assessment.**

Text Book:

1. Software Engineering (Third Edition), K K Aggarwal, Yogesh singh, New age International Publication (For unit 1,2,3,5 and case study of unit 4)
2. An integrated approach to software Engineering (Second Edition), Pankaj Jalote , Narosa Publishing House - (For Unit 4)

References:

1. Software Engineering (Seventh edition), Ian Sommerville – Addison Wesley
2. Software Engineering A practitioners approach (Sixth Edition), Roger S Pressman - Mc Graw Hill.
3. Fundamentals of Software Engineering (Second Edition), Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli - Pearson Education

5B09CSC Web Technology

Contact Hours per Week: 3
Credit : 3

Module -1: Introduction to internet and web, An overview of internet programming – WWW design issues. Introduction to HTML-structure of HTML, tags, attributes, syntax of tags ,starting and ending tags, html doc elements-<html>,<title>,<body>,physical style tags, listing, labeling, grouping, <img>-<a>

Module-2: Table tags-<tr>,<td>,<th> attributes-height, width, rowspan, colspan, border, color. Form-tag ,attributes-type-passwd,submit,radio,check,method,action.Frame-<frame>,<frameset>,<iframe>,<noframe> and other important tags and attributes.

Module-3 Javascript-datatypes, variables, function, object, array. Client-side object hierarchy and document. object Model,<script>,event handlers, javascript in urls. Windows and frames-dialog boxes, status line, navigator object, opening Windows, closing windows, Location object, history object.- Date object- math object- Accessing form object.

Module-4: Intro to PHP and advantages of ,PHP basics-functions, string, array, object, web techniques, database.

Module-5 Client-server model, introduction to cgi, environment variables, request-response model, encoding and decoding form data. Simple programming in CGI-database.

Books: 1. HTML-Definitive Guide O’reilley
2. Programming in PHP O’reilley
3.Programming in CGI O’reilley
4.Javascript-Definitive Guide O’reilley

Ref: 1.Complete reference in PHP-Steven Hozner
2.Beginning PHP5 (Wrox Programer)
3.Complete reference HTML-Tata McGraw Hill

5B10CSC Java Programming

Contact Hours per Week : 4 Theory
Credit : 4

Objectives: 1. Review OOPs concepts.
2. Learn features of java
3 Skill in java programming.

Module I Introduction to Java programming : Java technology; history; java as a new paradigm; features of java; Applications and applets (Simple examples); Java Development Kit Java Language fundamentals : Building blocks; Data types; variable declarations; wrapper classes; Operators and assignment; control structures; arrays; strings; String buffer classes.

Module II Java as an OOP Language: Defining classes; Modifiers; Packages;

Interfaces.

Module III Exception handling: Basics; handling exceptions in java; (Try, catch, finally, multiple catch, nested try, throw); Exception and inheritance; Throwing user defined exceptions; Advantages of exception handling.
Multithreading: Overview; Creating threads; thread life cycle; Priorities and scheduling; synchronization; Thread groups; communication of threads; Sample programs.

Module IV Files and I/O streams: Overview; Java I/O; file streams; FileInputStram and FileOutputStream; Filter Streams; RandomAccessFile; Serialization.
Applets : Introduction; Application vs. applets; Applet lifecycle; Working with Applets; The HTML APPLET tag; the java.Applet Package; Sample programs.

Module V The Abstract Window Toolkit:- Basic classes in AWT; Drawing with Graphics class; Class hierarchy; Event handling; AWT controls (Labels, Buttons, checkbox, radio buttons; choice control; list, textbox, scroll bars); Layout Managers.
JDBC Architecture; Working with JDBC; Processing Queries; The transactions Commit and Rollback; Accessing Metadata; ; Example Programs.

Text book :

1. Object Oriented Programming through JAVA, Radha Krishna, University Press.

Reference:

1. Programming with java: A primer, 3rd Edn; E. Balaguruswami; McGraw Hill
2. Java 2 The complete Reference, Schildt, McGraw Hill

5B12CSC Lab V (Java Programming)

Contact Hours per Week: 4
Credit : 2

5B13CSC Lab VI (Web Technology)

Contact Hours per Week: 4
Credit : 2

6B14CSC Data communication and Computer Network

Contact Hours per Week : 3
Credit : 3

Module I Introduction to data communication, important elements /components of data communication, Data transmission- Analog, Digital. Transmission media- Guided media, Unguided media. Synchronous / Asynchronous data transmission. Line configuration – Simplex, Half duplex, Duplex. Network topologies – star, Bus, ring, Mesh.

Computer networks, Use, network hardware, network structure- point to point connection, multicast, broadcast, classification of networks-LAN, WAN, Man. Network software – protocol hierarchies.desigh issues for layers, interfaces and services- connection oriented, connection less.

Module II Reference models, the OSI reference model, TCP / IP reference model. Comparison between OSI and TCP / Ip models.
Data Link Layer , Design issues, Services to network layer, Framing- charater count, character stuffing, bit stuffing, physical layer coding violation. Error control, flow control, Elementary data link protocols- unrestricted simplex protocol, simplex stop and wait protocol, simplex protocol for a noisy channel.

Module III Network layer, design issues, services to the transport layer, routing algorithms- adaptive, non adaptive algorithms, optimality principle, dijkstras shortest path routing algorithm, flow based routing, hierarchical routing, congestion control algorithms – the leaky bucket algorithm, the token bucket algorithm.

Module IV Transport layer, design issues, connection management-addressing, establishing and releasing connection, transport layer protocols- TCP, UDP.

Module V Application layer, network security, traditional cryptography, substitution ciphers, transposition ciphers, fundamental principles, secret key algorithm, data encryption standard, DES chaining, DES breaking. Public key algorithm, RSA algorithm.

References

1. B Forousan, Introduction to data communication and networking
2. A S Tanenbaum . Computer Networks.
3. Data communication and Networks, Achyut S. godbole, TMH
4. Computer Networks – fundamentals and Applications, Rajesh, Easearakumar & Balasubramaian, Vikas pub.

6B15CSC Computer Organization

Contact Hours per Week : 3Theory
Credit : 3

Module I Basic structure of computer-Types of computers-Functional Units-Basic operational Concepts-Bus structure-Multiprocessors and Multi computers-Data representation-Fixed Point representation and floating Point representation.

Module II Register Transfer and Micro operations – Register Transfer language-Register Transfer-Bus and memory Transfer-Three state bus buffers-Memory Transfer-Basic Computer Organization and Design – Instruction Codes – Fetch & Decode Instructions –

Register Reference Instructions – Memory Reference Instruction – Input output & Interrupt.

Module III Micro Programmed Control – Control Memory – Address sequencing – Central Processing Unit – General Register Organization – Control word – Stack Organization – Register stack - Memory Stack – Reverse Polish notation – Evolution of Arithmetic expressions – Instruction Formats – Addressing modes – Data Transfer and Manipulations – reduced Instruction set computer(RISC)

Module IV Input Output Organization – Peripheral Devices – Input/Output Interfaces – Asynchronous Data Transfer – Modes of transfer –Priority Interrupt – Direct Memory Access (DMA) - Input Output Processor - Serial Communications.

Module V Memory Organization – Hierarchy – Main memory – Auxiliary Memory – Associative Memory – Cache memory – Mapping – Multiprocessors – Characteristics of multiprocessors - Inter connection structures – Inter Processor Arbitration.

Text Books
1. Computer system Architecture –M.Morris Mano - PHI Pvt Limited
2. Computer Organization - Carl Hamacher –International Edition

References
1. Computer Organization and Architecture , William Stallings, 7th Edn, Pearson Education.
2. Computer Architecture & Organization John P Hayes –Mc Graw Hill

6B16CSC Operating Systems

Contact Hours per Week : 3 Theory
Credit : 3

Objectives:
1. Introduce basic concepts of operating systems.
2. Familiarize with features of operating systems.
3. Basics of design of operating systems.
4. Overview of Linux.

Module I Concepts – Importance – Resource manager – Views – Design considerations – I/O programming – Interrupt structure and processing. (Text Book 1)
Batch Processing System – Multi programming system - Time Sharing System – Real Time System. (Text book 2).

Module II Processor management: Process – interacting processes - Threads – Scheduling policies – job scheduling – process scheduling – Multi processor OS. Dead locks – Dead lock handling techniques. (Text book 2).

Module III Memory management: Single contiguous allocation – Partitioned allocation – Relocatable partitioned – Paging – Demand paging – Segmentation – Segmentation and demand paging – Other schemes (Text book 1).

Module IV Device management: Techniques – Channels and control units – I/O traffic controller, I/O scheduler, I/O device handlers – Virtual devices
Information management: Introduction – General model - SFS – BFS – ACV – LFS – PFS – ASM . (Text book 1).

Module V

Unix and Linux – History; over view; Process, memory management – I/O – file system – security. (Text Book 3)

Text Book

1. Stuart E Madnick and John J Donovan, "Operating Systems", Tata McGraw-Hill, 2005
2. Dhamdhere, "Systems Programming and Operating Systems", 2nd Revised Edn, TMH
3. A. S. Tanenbaum, "Modern Operating systems"; PHI

5B11CSC - E01 Algorithm Analysis and Design

Contact Hours per Week : 3 Theory + 2 Practical

Credit : 4

Module 1: Introduction- Definition of algorithm, Areas of algorithm study, performance analysis, Time and space complexity, asymptotic notations (O, Ω, Θ).

Module 2: Divide and Conquer – general method, Binary search, Finding the maximum and minimum, Merge sort, Quick sort, Performance measurement of quick sort, selection, Strassen's matrix multiplication.

Module 3: Greedy method – General method, knapsack problem, job sequencing with dead lines, minimum cost spanning trees, prims algorithm, kruskals algoritms, optimal merge patterns, single source shortest path, optimal binary search trees.

Module 4: Dynamic programming – General method, multistage graph, allpairs shortest path, single shortest path, 0/1 knapsack travelling sales person problem.

Module 5*:* Basic traversal and Search techniques – Breadth First Search and traversal, Depth First Search and Traversal, Bi-connected components and DFS; Backtracking – General methods, 8-queens problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

Text:

1. Ellis Horowitz, Sartaj Sahni, S Rajasekharan – Computer Algorithms/C++ - Second Edition, Universities press, 2008 (Paperback Edn)

Reference:

1. Introduction to the design and Analysis of Algorithms, Anany Levitin, 2nd Edn, pearson education.

5B11CSC - E02 Network Programming

Contact Hours per Week : 3 theory + 2 Practical

Credit : 4

Module I Introduction – A Simple Day Time Client – Protocol Independence – Error Handling – A Simple - Day Time Server
The Transport Layer : TCP, UDP – TCP Connection Establishment and Termination – TIME_WAIT State – Port Numbers – Concurrent Servers – Buffer Size and Limitations – Standard Internet Services – Protocol Usage by Common Internet Applications

Module II Socket Introduction – Socket address Structures – Byte Ordering Functions – Byte Manipulation Functions – Elementary TCP Sockets – socket , connect, bind, listen, accept, fork and exec, close, getsockname and getpeername functions

Module III TCP Client/Server Example – TCP Echo Server - main(), str_echo() – TCP Echo Client - main(), str_cli() – startup – termination – Shutdown of Server Host

Module IV Socket Options – getsockopt and setsockopt functions – Socket States – Generic Socket Options – TCP Socket Options

Module V Name and Address Conversions - DNS – gethostbyname – gethostbyaddr – getservbyname – getservbyport – getaddrinfo – freeaddrinfo – host_serv – tcp_connect – tcp_listen functions

Text Book

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, “Unix Network Programming The Sockets Networking API Volume I”, Pearson Education

Reference

1. Barry Nance, “Network Programming in C” , Prentice Hall

5B11CSC - E03 Soft Computing

Contact Hours per Week : 3 theory + 2 Practical
Credit : 4

Module I Basics of Artificial Neural Networks – characteristics of Neural networks, Historical development of NN, ANN Terminology, Models of neuron. Topology, basic Learning Laws, Activation dynamic Models, symaptic Dynamic Models, Learning methods.

Module II Pattern recognition problem, Basic Functional Units, Pattern Recognition Tasks by functional units, feed forward Neural networks – introduction, analysis of pattern Association networks, Analysis of pattern classification Networks, Analysis of Pattern mapping networks.

Module III Feed-back neural networks – Introduction, analysis of Linear Auto-associative feed forward networks, analysis of pattern storage networks, applications of ANN.

Module IV Introduction to classical sets – properties, operations and relations; fuzzy sets, membership. Uncertainty, operations, Properties, Fuzzy relations, cardinalities, membership functions.

Module V Fuzzification, membership values assignment, development of rule base and decision making system, Defuzzification to crisp sets, Defuzzification methods.

Text book :

1. "Artificial Neural Networks", Yegnanarayana, PHI.
2. "Neural Networks, Fuzzy Logic , Genetic algorithms : synthesis and Applications", Rajasekharan and Rai, PHI.

Reference :

3. "Neural networks", James A freeman & Davis Skapura, Pearson education, 2002.
4. " Neural Networks", simon Hakins, Pearson education.
5. "Neural networks and fuzzy Logic system", Bart Kosko, PHI Publications.

5B11CSC - E04 Numerical Methods

Contact Hours per Week :3 Theory + 2 Practical
Credit : 4

Module I Introduction to Numerical methods: Nature of numerical problems; computer based solutions; number representations; notions of accuracy, convergence, efficiency, complexity; solutions of Non-linear equations : Bisection method; Regula-Falsi; Newton Raphson; secant; Successive approximation method.

Module II Interpolation techniques: Linear interpolation; Newton's forward and backward formulae; Lagrange's interpolation; Bessel Functions; Linear Regression; Cubic splines; Chebyshev Polynomial.

Module III Concept of differentiation and integration; Graphical Interpretation; Cubic Spline based Numeric differentiation; Numeric integration;: trapezoidal, Simpson's Romberg, Gaussian and Filon's methods.

Module IV Matrix based solutions of simultaneous linear equations: Gauss Jordan Method; Gauss elimination with Back – substitution; LU decomposition method.

Module V Differential equations: Picard's method; Euler's and modified Euler's method; Runge-Kutta; Predictor – Corrector methods; partial differential Equations; Jacobi and Gauss-Siedel methods.

Text books: 1. Numerical Techniques in C, Kameshwar, BPB Publications.
2.Computer Oriented Numerical Methods, Datta, Vikas.
3.Computer Oriented numerical Methods, Rajaraman V., 3rd Edn, PHI

5B11CSC - E05 Computer Graphics

Contact Hours per Week : 3 Theory + 2 practical
Credit : 4

Objectives :

i. Introduce basics of Computer Graphics.
ii. Basic concepts in 2D and 3D graphics.
iii. Review of selected fundamental algorithms in Graphics.
iv. Skill in writing Graphic programs.

Module I Introduction, Overview of Graphics Systems, Display devices, Input devices, Hard-Copy devices, Graphics software.

Module II Line Drawing Algorithms-DDA, Bresenham, Circle Generating Algorithm – Midpoint Algorithm, Area filling algorithms – Flood Fill and Boundary Fill algorithms.

Module III Output primitives-Color and Grayscale levels, 2D Transformations-Translation, Rotation, Scaling, Reflection, Shear, Matrix Representation and Homogenous Coordinates, Composite Transformations.

Module IV Two Dimensional viewing, Window-to-viewport Transformation, Clipping - Point Clipping, Line Clipping – Cohen Sutherland Algorithm, Polygon Clipping – Sutherland Hodgeman Algorithm, Text clipping.

Module V 3D object representations-Polygon surfaces, Polygon tables, Plane equations, Polygon Meshes, 3D transformations-Translation, Rotation, Scaling, Rotation about an arbitrary axis, Reflection, Shear, 3D viewing- Parallel Projection, Perspective Projection.

Text Book

1. Donald Hearn and M.Pauline Baker, "Computer Graphics-C Version", Second Edition, Pearson Education, 2005.
2.

References

1. Foley, Vandam, Feiner, Huges, "Computer Graphics: Principles & Practice", Second edition in C, Pearson Education, 2005
2. Ranjan Parekh, "Principles of Multimedia", ,Tata McgrawHill,2006
3. D.P. Mukherjee, "Fundamentals of Computer Graphics and Multimedia", PHI.
4. "Procedural elements of Computer Graphics", Rogers, Mc-Graw Hill.
5. "Mathematical elements of Computer Graphics", Rogers, Mc-Graw Hill.
6. Steven Harrington, "Computer Graphics- A Programming Approach", Second Edition, McgrawHill International.

5B11CSC – E06 Digital systems and Microprocessors.

Contact Hours per Week : 3 Theory + 2 Practical
Credit : 4

Aim : To introduce Microprocessors through 8085 and 8086.

Objectives :

1. Familiarize with basics of digital electronics.
2. Familiarize with 8086 architecture.
3. Skill in writing assembly language programs.
4. Understand Interrupts and DMA techniques.

Module I Digital Logic: Basic Gates; Boolean algebra; NOR and NAND gates; AND-OR-INVERT gates; positive and Negative logic.

Combinational Logic Circuits: Boolean laws and Theorems; SOP methods; Truth table and K-map, K-map simplification (up to four variable);
Data-processing circuits (Definition / function and one example each) : Multiplexers; Demultiplexers; decoders; Encoders; Ex-Or gates; Parity generators and checkers.
Arithmetic Building blocks; ADDER – SUBTRACTOR.

Module II FLIP FLOPS : RS ; Gated FFs; Edge triggered RS, D and JK flip Flops; Flip flop Timings; JK Master Slave Flip flops.
Registers : Types : Serial in – Serial Out; Serial In – parallel out; parallel In – Serial Out; Parallel In – parallel Out; ring Counters.
Counters : Asynchronous counters ; Decoding gates; Synchronous counters; changing the counter Modulus; Decade counters.

Module III Introduction to 16-bit microprocessor – 8086, Architecture of 8086, Functional Block Diagram, Register Organization of 8086, Signal Description of 8086, Physical Memory Organization, Memory Mapped and I/O Mapped Organization, General Bus Operation, I/O Addressing Capability, Minimum and Maximum Mode 8086 System and Timings.

Module IV Addressing Modes of 8086, Machine Language Instruction Format, Assembly Language Programming of 8086, Instruction Set of 8086-
(Selected Data transfer instructions, Arithmetic and Logic instructions, Branch instructions, Loop instructions, Processor Control instructions, Flag Manipulation instructions, Shift and Rotate instructions, String instructions, Assembler Directives and operators. Example Assemble language program..)**
Introduction to Stack, STACK Structure of 8086, Interrupts and Interrupt Service Routines, Interrupt Cycle of 8086, Non-Maskable and Maskable Interrupts.

Module V Data transfer schemes – Programmed IO, Interrupt driven IO and DMA.
Programmable Peripheral Interface 8255, DMA Controller 8257, Programmable Interrupt Controller 8259A,
[General Features of Intel Processors - Intel 80286, 80386, 80486, Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV and Itanium.]**

*** Should not be included for end semester Examination, however shall be included for internal assessment*

Text Book

1. Advanced Microprocessors and Peripherals – Architecture, Programming and Interfacing by A.K. Ray and K.M. Bhurchand, Tata McGraw Hill,2002 Edition
2.Digital Principles and Applications; Leach and Malvino; TMH; 6th edn

Reference Books

1. Microprocessors and Interfacing – Programming and Hardware by Douglas V Hall, 2nd Edition, Tata McGraw Hill, 2002.
2. The Intel Microprocessors 8086/8088, 80816/80188, 80286, 80486 Pentium and Pentium Pro Processor – Architecture, Programming and interfacing by Barry B Brey, 4th Edition, PHI
3. Microprocessor x86 Programming by K.R. Venugopal and Raj Kumar – BPB publications
4. Microcomputer Systems – The 8086/8088 Family Architecture, Programming & Design by Yu Cheng Liu, Glenn A Gibson – PHI Edition.

6B17CSC – E07 Information Security

Contact Hours per Week : 4 Theory
Credit : 4

Module I Introduction to Information Security- The need for Security, Principles of security - confidentiality, Authentications, Integrity, Non-repudiation. Types of attacks- Passive attacks, Active attacks, Vius, Worm, Trojan horse. Introduction to Cryptography, Steganography, Secret Sharing.

Module II **Symmetric Key Encipherment**:- Traditional symmetric Key Ciphers: Introduction- Kirchhoff's principle, cryptanalysis, categories of traditional ciphers; Substitution Ciphers - monoalphabetic ciphers, polyalphabetic ciphers; Transposition Ciphers - keyless and keyed transposition ciphers, Stream and Black Ciphers - stream ciphers, block ciphers.

Module III **DES(Data Encryption Standard):-** Introduction, DES Structure - initial and final permutations, rounds, cipher and reverse cipher, examples; DES Analysis - properties, design criteria, DES weaknesses; Multiple DES - double DES, triple DES; Security of DES - brute-force attack, differential cryptanalysis, linear cryptanalysis.

Module IV **Public key Cryptosystem**: Principles of Public Key Cryptosystems- Public Key Cryptosystem, Applications of Key Cryptosystems, Requirement for Public Key Cryptosystem, Public Key Cryptanalysis. RSA Algorithm – Description of the Algorithm, Computational Aspects, Security of RSA

Module V **Digital Signature:-** Comparison- inclusion, verification method, relationship, duplicity; Process- needs for keys, signing the digest; Service- message authentication, message integrity, nonrepudiation, confidentiality; Attacks on Digital Signature- attack types; Digital Signature Schemes- RSA digital signature schemes.

Text Books:

1. Cryptography and Network Security", Behrouz A Forouzan, Tata McGraw-Hill Publishing Company Limited, Special Indian Edition 2007. (For Module - I, II, III, V).

2. Cryptography and Network Security Principles and Practices, Willian Stalling, Pearson Education (For Module - IV).

Reference Text

1. Fundamentals of computer security, Josef Pieprzyk, Thomas hardjino and Jennifer Sebberry, Springer International Edition 2008.

2.

6B17CSC – E08 Mobile Communications

Contact Hours per Week : 4 Theory
Credit : 4

Module I Introduction – history of wireless communication, A simplified reference model, frequencies for radio transmission, signals, Antennas, signal Propagation, Spread spectrum – DSSS and FHSS, Cellular systems.

Module II SDMA, FDMA, TDMA and CDMA, GSM – Mobile services, system Architecture, Radio interface, Protocols, Localization and Calling, Handover, Security, GPRS.

Module III
Wireless LAN – infrared versus Radio transmission, IEEE 802.11 – system Architecture, Protocol architecture, Physical Layer, MAC Layer, MAC Management, 802.11b, 802.11a. Introduction to Bluetooth – IEEE 802.15.

Module IV
Mobile IP – entities and Terminology, IP Packet delivery, Agent discovery, Registration, tunneling, IPV6, Introduction to MANET, TCP over 2.5/3G Wireless Networks.

Module V WAP (1.x) – Architecture, Wireless Datagram Protocol, Wireless Transport Layer security. Wireless Transaction Protocol, wireless Session Protocol, wireless Application Environment, wireless Markup Language, WML script, Introduction to WAP 2.0.

Text book :

1. Mobile communications, Jochen Schiller, 2nd edn, Pearson education.

Reference :

1. Wireless Communication Technology, R. Blake, Thomson Delmar, 2003.
2. Mobile communication engineering: theory and Applications, W. C. Y. Lee, 2nd edn, Mc Graw Hill international Edn, 1998.
3. Wireless digital Communication, Feher, PHI, 199.

4. Principles and Applications of GSM, Vijay K. garg & J. e. Wilkes, Prentice Hall, 1999

6B17CSC – E09 Data Mining

Contact Hours per Week : 4 theory
Credit : 4

Module I Introduction; data warehousing – what is, Multidimensional data model, OLAP operations, warehouse schema, Data warehousing Architecture, warehouse server, Metadata, OLAP engine, data warehouse Backend Process.

Module II Data mining – what is, KDD vs data mining, DBMS vs data mining, DM Techniques, issues and challenges, Applications. (Case studies) *

Module III Association rules – What is, Methods, a priori algorithm, partition algorithm, Pincer- search algorithm, FP-tree growth algorithm, incremental and Border algorithms, Generalized Association rule.

Module IV Clustering techniques – Paradigms, Partitioning Algorithms, k – Medoid algorithms, CLARA, CLARANS, hierarchical clustering, DBSCAN, Categorical Clustering, STIRR.

Module V Decision trees – what is, tree construction principles, Best split, Splitting indices, Splitting criteria, decision tree construction algorithms, CART, ID3, C4.5, CHAID.
Introduction to web, spatial and temporal data mioning.

Text book :

1. Data mining techniques, A K Pujari, University press.

Reference :

1. J. Han, M. Kamber, “Data Mining Concepts and Techniques”, Harcourt India Pvt Ltd.
2. M. Dunham, “ Data Mining : introductory and Advanced Topics”, Pearson Pub.

6B17CSC – E10 C# and .NET Frame Work

Contact Hours per Week : 4
Credit : 4

Module – 1: Introduction to C# Introducing C#, Understanding .NET, Overview of C#, Literals, Variables, Data types, Expressions, Branching, Looping, Methods, Arrays, Strings, Structures, Enumerations

Module– 2: Object oriented aspects of C# Classes, Objects, Inheritance, Polymorphism, Interfaces, Operator Overloading, Delegates, Events, Errors and Exception

Module– 3: Application Development on .NET Building Windows Applications, Accessing Data with ADO.NET

Module – 4: Web Based Application Development on .NET Programming Web applications with Web Forms, Programming Web Services

Module – 5: The CLR and the .NET Framework Assemblies, Versioning, Attributes, Reflection, Viewing Meta Data, Type Discovery, Reflecting on a type, Marshalling, Remoting, Understanding Server Object Types, Specifying a server with an Interface, Building a server, Building the Client, Using Single Call, Threads.

Text Books:

1.Programming in C#, E.Balagurusamy (Unit I, II)

2.Prgramming in C#, J. Liberty 2nd Edition – O'Reilly (Unit III, IV, V)

6B17CSC – E11 Digital Image Processing

Contact Hours per Week : 4
Credit : 4

Module I Images – DIP components – Problems and Applications – motivation and perceptive – Operations – Imaging – electronic camera – Human Eye – 3D imaging – Depth from triangulation , time-of-flight, interferometery, shading, tomography, Sampling – quantization, Colour Image representation, Volumetric data.

Module II Images in Java – java2D API – java advanced imaging – image manipulation – storage – reading and writing images – display – printing – pixel processing – gray level and colour enhancement – mapping – image histogram – Histogram equalization – Colour processing.

Module III Neighbourhood operations – convolutions and correlation – Linear and rank filtering – Edge detection – Hybrid adaptive fileters – frequency domain – spatial frequency – fourier theory – DFT – investigating spectra – image filtering – deconvolution.

Module IV Geomteric operation – simple techniques – Affine transformations – Algorithm – interpolation schemes – Wrapping and morphing – segmentation – thresholding – Contextual techniques.

Module V Morphological image processing – Basic concepts – operations – Morphological filtering – Morphological algorithms – Gray scale morphology – image compression. Redundancy – Performance characterization – Lossy and lossless compression techniques – compression of moving images.

Text book: Digital image Processing : A practical introduction using Java ; Nick Efford; Pearson Edn.

Reference:

1. Digital Image Processing; Gonzalez and Woods; Pearson Edn.
2. Digital image Processing; B. Jahne; Springer international Edn.

6B17CSC – E12 Data Compression

Contact Hours per Week : 4

Credit : 4

Module I Data Compression Lexicon : Introduction to data compression – Dawn age – coding – Modeling – Ziv and Lampel lossy compression.

Minimum redundancy coding (the Dawn age) : tha shanon – fano algorithm – The Huffman algorithm – into the Huffman code : counting the symbols, buiding the tree – compression code.

Module II Adaptive Huffman code : adaptive coding - Updating the Huffman tree – The code.

Arithmetic Huffman coding : Arithmetic coding – the code.

Module III Statistical modeling : Higher order modeling – Finite context modeling – adaptive modeling – highest order modeling.

Dictionary – based compression : static vs adaptive – Israeli roots – ARC.

Sliding window compression : The algorithm – LZSS Compression – the code – Conpression code.

Module IV LZ78Compressiojn – Decompression.

Speech compression : Digital audio Concepts – Lossless compression of sound.

Module V

Video compression – JPEG compression – implementing DCT – Complete code listing.

Reference:

1. Mark Nelson; "the data Compression Book", BPB 2003
2. Khalid Sayood, "Introduction to Data compression", Morgan Kaufman, 2003
3. Yun Q Shi, Huifang sun; "Image and Video Compression for multimedia Engineering", CRC Press, 2008.
4. David S. Tanbman and Michael W Marcellin, "JPEG – 2000 Image Compression fundamentals, Standard Practice", Kluwer Academic, 2002.

6B17CSC – E13 Linux System Administration

Contact Hours per Week : 3 Theory + 1 practical
Credit : 4

Module I Features and benefits of Linux- basic concepts of multi user system-open source, freedom-Linux-components of Linux, types of users in Linux, types of files.

Introduction-login, password, creating an account, shell and commands, logout,changing password-files and directories-pathname-directory tree-currend working directory-referring home directory-cbeating new directories,copying files,moving files,deleting files and directories-types of shell-wild cards-hidden files- looking at files: cat, more-online help:man.

Module II Vi editor-different modes-command mode, insert mode, last line mode-redirecting input/output-filter, pipes, file permissions, user, group, changing file permissions - mounting floppy,HDD, CDROM-file systems-structure of /etc/fstab-Bourne shell scripts: script execution-variables and parameters, if, for, case, while constructs.

Module III Linux Administration: Introduction-various parts of the OS-kernel, system program, application program, system calls-important parts of the kernel
Boot procesS: booting-LILO boot process,/edc/lilo.conf, GRUB, /etc/grub.conf-runlevels-GUI,X windows- rc files, startup scripts.

Module IV Major services in linux system : init,/etc/inittab file -login from terminal3, syslog-periodic command execution: at and cron, crontab file
System configuration files:/etc/sysconfig/.....files,keyboard,mouse etc
System security: password,/etc/passwd file-shadow password,/etc/shadow-file permissions, chmod and umask-adding and deleting users-host security, tcp wrappers,/etc/host.allow, /etc/host.deny.

Module V System Maintance: tmpwatch-logrotate-basic system backup and restore operation-Basic shell configuration for bourne and bash shell : /etc/profile,~/.bashrc,~/.bash_profile.
Linux Installalation : Partitioning, MBR, SWAP, filesystem managing-different packages, rpm-installation of packages-starting and stopping different services.
** Comparative study of major features of Linux and windows.

Reference:

1. Unix in a nutshell,by Daniel Gilly, O'Reilly & Associates
2. Unix Shell Programming, Yeshwanth kanethkar
3. Linux Administration handbook, Nemeth, PHI
4. Essential System Administration, O'reilly & Associates.
5. Red Hat linux Bible
6. A user guide to the unix system, Thomas,Yates Tata McGraw Hill

Sd/-
Dr.G.Raju,
Chairman,BOS Computer Science(UG)

KANNUR UNIVERSITY

Course Structure
and
Syllabus

FOR

Computer Science (Complementary)

UNDER

CHOICE BASED CREDIT SEMESTER SYSTEM

w.e.f 2009 ADMISSION

Scheme Complementary (Computer Science)

No	Semester	Course Code	Title of the course	Hours / Week	Credits
1	I	1C01CSC	Introduction to IT and C programming	2	2
2	I	1C02CSC	Lab – I (C Programming)	2	1
3	II	2C03CSC	Programming in C++	2	2
4	II	2C04CSC	Lab – II (C++ Programming)	2	1

5	III	3C05CSC	Data Base Management System	3	2
6	III	3C06CSC	Lab III (DBMS Lab)	2	1
7	IV	4C07CSC	Visual Basic	3	2
8	IV	4C08CSC	Lab IV (Visual Basic)	2	1

1C01CSC Introduction to IT and C Programming

Contact Hours per Week : 2 Theory
Credit : 2

Module I Information technology: introduction to computer, different classification of computer, I/O devices, commonly used I/O devices Key board, mouse, VDU, different type of printers, Memory organization, primary memory, RAM, ROM, PROM,EPROM, cache memory, secondary storage device, Floppy disk, Hard disk, CD, DVD, etc.

Module II Representation of information: number system, binary, octal, hexadecimal system, different code used BCD, ASCII,EBCDIC, GRAY Code,computer languages: machine language, assembly language, high level language , compilers , interpreter. Problem solving using computers algorithm and flow charts examples.

Module III The C character set, Identifiers and keywords, Classes of Data Types, constants, variable declarations. Expressions, statements, symbolic constants, operators and expressions: arithmetic operators, unary operators, relational operator, logical operators, assignment operator, the conditional operator. Library functions: data input and output functions like getchar(), putchar(),scanf(), printf(), gets and puts. Control statements: Branching: The if-else statements. Looping: The while, do-while and for loops. The switch statements, Break and continue, comma operator.

Module IV Functions**,**Defining a function, accessing a function, function prototype, passing arguments to a function, Returning from a function, recursion, program structure. Storage classes: automatic, static, register and extern(global). Multi file program.

Module V Arrays, Structure and Union : Defining an array, processing an array, passing arrays to functions, multidimensional arrays. Character arrays and strings. Structure and union. Defining a structure, processing a structure. Passing structure to functions, union. Concepts of pointers: Pointers to built in data types only.

Text Book :

2. ANSI C, E. Balagurusamy, 3rd edition McGraw-Hill Publication

Reference books:

5. Computer Basics and c Programming, V. Rajaraman, PHI, 2008
6. Programming with ANSI and Turbo C, Ashok N. Kamthane, 1st edn, Pearson Education.
7. Let us C, Yeshvanth Kanethkar, 3rd Edn, BPB,
8. Programming with C in Linux, NIIT, PHI.
9. C by Example, Noel Kalicharan, Cambridge University press.

1C02CSC Lab I (C Programming)

Contact Hours per Week : 2 Practical
Credit : 1

List of Programs

Students have to record a minimum of 20 programs.

15. Develop a program that reads a floating point number and then displays the right most digit of the integral part of the number.
16. The straight line method of computing the yearly depreciation of the value of an item is given by

$$Depreciation = \frac{purchase\ price - Salvage\ value}{Year\ of\ Service}$$

Develop a program to determine the salvage value of an item when the purchase price, year of service and annual depreciation are given.

17. Develop an interactive program to demonstrate the process of multiplication. The program should ask the user to enter two two-digit integers and print the product of integers as shown below:

```
                          45
                    X     37
                    __________
   7 x 45 is                    315
   3 x 45 is                   135
                    __________
   Add them             1665
                    __________
```

18. Develop a program to find the number of and sum of all integers grater than 100 and less than 200 that are divisible by 7.
19. A set o two linear equations with two unknowns x1 and x2 is given below:

$$a\,x_1 + b\,x_2 = m$$
$$c\,x_1 + d\,x_2 = n$$

The set has a unique solution

$$x1 = \frac{md-bn}{ad-cb} \qquad x2 = \frac{na-mc}{ad-cb}$$

Provided the denominator ad – cb is not equal to zero.

Develop a program that will read the values of the constants a, b, c, m and n and compute the values of x1 and x2. An appropriate message should be printed if ad – cb = 0.

20. Admission to a professional course is subject to the following conditions:

e. Marks in mathematics >=60.

f. Marks in Physics >=50.

g. Marks in Chemistry >=40.

h. Total in all three subjects >=200 Or
Total in Mathematics and Physics >=150.

Given the marks in the three subjects, develop a program to print whether an applicant is eligible or not.

21. A cloth showroom has announced the following seasonal discounts on purchase of items:

Purchase amount	Discount	
	Mill cloth	Handloom
Items		
0 – 100	--	5%
101 – 200	5%	7.5%
201 – 300	7.5%	10%
Above 300	10%	15%

Develop a program using switch and if statements to compute the net amount to be paid by a customer.

22. Develop a program that will read the value of x and evaluate the following function :

$$y = \begin{cases} 1 \; for \; x < 0 \\ 0 \; for \; x = 0 \\ -1 \; for \; x < 0 \end{cases}$$ using nested if, else if and

conditional operator.

23. Develop a program using do—while loop to print the first m Fibonacci numbers.

24. Develop a program to print the following output using for loop :

```
*******
******
*****
****
***
**
*
```

11. Develop a program to read the age of 100 persons and count the number of persons in the age group 50 to 60. Use for and continue statements.

12. Develop a program to read a positive integer and print its binary equivalent.

13. Develop a program to compute Euler's number e using the following formulae :

E = 1+1/1! +1/2!+1/3! + ….+1/n!

14. Develop a program to evaluate the following function to 0.0001% accuracy.
    $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots.$
15. Develop a program to sort a list of n positive integers in ascending order.
16. Develop a program to search a list of integers for a key k. (Sequential search).
17. Given two one dimensional sorted (ascending) arrays A and B. Develop a program to merge them into a single sorted array C that contains every item from arrays A and B, in ascending order.
18. Develop a program for matrix multiplication.
19. Write a program for fitting a straight line through a set of points (x_i, y_i), i=1,2,…n. The straight line equation is y = mx +c. the values of m and c are given by:

    $$m = \frac{n\sum(x_i y_i) - (\sum x_i)(\sum y_i)}{n(\sum x_i^2) - (\sum x_i)^2} \qquad c = \frac{1}{n}(\sum y_i - m\sum x_i)$$

18. Develop a program to read a string and determine whether it is a palindrome.
19. Develop a program to read a text and count occurrences of a particular word.
20. Develop a program to replace a particular word by another word in a given string.
21. Develop a program to read a set of n names and sort them in alphabetic order.
22. Develop a function prime that returns 1 if its argument is a prime number and returns a zero otherwise. Develop a main program to read n integers into an array and print all prime numbers in the array.
23. Develop a recursive function to find factorial of a number.
24. Develop a function that will scan a character string passed as an argument and convert all lower case characters into their uppercase equivalent.
25. Design and code an interactive modular program that will use functions to read a matrix of m by n size, compute column averages and row averages, and then print the entire matrix with averages shown in respective rows and columns.
26. Define a structure data type called time-struct containing three members integer hour, integer minute and integer second. Develop a program that would assign values to the individual members and display the time in the following form: 16:30:21
27. Define a structure named census with the following thre members :
    - A character array city[] to store names.
    - A long integer to store population of the city.
    - A float member to store the literacy level.

    Develop a program to do the following:
    - To read details for 5 cities.
    - To sort the list of cities in alphabetic order.
    - To sort the list based on population.
    - To display the sorted lists.
28. Develop a function (using pointers) that reverses the elements of a given array.
29. Develop a program to copy the contents of one file into another.
30. Two files DATA1 and DATA2 contain sorted lists of integers. Develop a program to produce a third file DATA which holds a single sorted, merged

list of these two lists. Use command line arguments to specify the file names.

2C03CSC Programming in C++

Contact Hours per Week : 2 Theory
Credit : 2

Module 1 C++ Introduction,object-oriented programming, advantage of OOp, characteristics of Object oriented Languages. Objects, Classes, Inheritance, Reusability, creating new data types, polymorphism, abstraction, encapsulation, Data hiding, message communication. **Programming Basics**: Basic program construction, Input and Output using cin and cout. Preprocessor directives, Comments. Data types, Manipulators, Type conversions. Operators and expressions. Library functions. Control Constructs: Branching and Looping. Switch and go to statements. Break and continue.

Module II Structures and Functions Structure, A simple Stricture, Specifying the structure, Defining a structure, Accessing Structure members, Enumerated data types. Functions-simple functions.passing arguments to functions.Returning values from functions.reference arguments. Overloaded functions.Inline functions.default arguments. Variables and storage classes. Returning by reference.

Module III Classes and Arrays. Classes and Objects, C++ objects as physical objects. C++ objects as data types. Constructors, Objects as function arguments, Returning objects from functions. Array Fundamentals, Arrays as Class member data. Arrays of objects, Strings.Address and Pointers. Pointer and Arrays, pointers and functions, pointers and strings, memory management: new and delete, Pointers to Objects.

Module IV Operator Overloading and Inheritance Overloading unary operators, overloading binary operators, data conversion, pitfalls of operator Overloading and conversion.

Inheritance- Derived class and base class. Derived class' constructors, overriding member functions. Public and private inheritance. Types of Inheritance. Container ship: class within class.

Module V Virtual functions. Static functions, Friend functions. Assignment and copy initialization. This pointer Files and Streams: Streams: String I/O, character I/O ,Object I/O, I/O with multiple objects. File pointers. Disk I/O with member functions. error handling. Redirection, commandline arguments, .

Text book :

1.Object Oriented Programming in C++, Robert Lafore, Techmedia

Reference Books: .

1. Object Oriented Programming with C++; E. Balagurusamy; 3rd Edn; TMH 2006
2. Programming in C++, M.T. Somashekara, Prentice Hall of India, New Delhi
3. Object Oriented Programming with ANSI & Turbo C++, Ashok N. Kamthane, Pearson Education
4. Let us C++, Yeshwanth Kanethkar, BPB

2C04CSC Lab II (Programming in C++)

Contact Hours per Week : 2
Credit : 1

List of Programs

Students have to do and record a Minimum 20 programs
All programs must use OOPs concept(s)

1. Program to find the factorial of a number using recursion.
2. Program to find whether the given number belongs to fibonacci series.
3. Program to find whether the string is palindrome or not. Use pointers.
4. Write a program to sort n numbers.
5. Program to find biggest, smallest, sum and difference of two numbers using inline function.
6. Program to find the area and volume of respective figures using function overloading.
7. Program to add one day to a given date.
8. Program to add and subtract two matrices.
9. Program to multiply two matrices.
10. Program to find the trace and transpose of a matrix.
11. Program to show stack operations.
12. Create a class time comprises hr,min and sec.as member data and add() and display() as member functions.Use constructor to initialise the object. write a main function to add two time objects, store it in another time object and display the resultant time
13. Program to negate the elements of an array. Use operator overloading function with the operator -.
14. Program to compare two strings. Use operator overloading (==). Do not use any built in functions.
15. Define a class POLAR which comprises polar coordinates like radius and angle as member data. Design another class RECTANGLE comprises rectangular coordinates like x and y. Use data conversion functions to convert from rectangle to polar coordiantes and vice versa. You need to use the following trigonometric formulae:

    x= r*cos(a);

    y= r*sin(a);

    a= atan(x/y);

    r= sqrt(x*x + y*y);
16. Define a class student with name, reg.no, date of birth and name of college as member data and functions to get and display these details. Design another class Test with subjects of study and grade for each subject as member data and corresponding input and output functions. Derive a class Result from both Student and Test classes and Print the Result of each student with relevant information.
17. Start with an array of pointers to strings representing the days of the week. Provide functions to sort the strings into alphabetical order. Use pointers
18. Create a class person with personal details. Define two functions, setdetails and

printdetails. Declare array of pointers to person class and write a main function to set and print the details of n persons using pointers.

19. Design two classes A and B with member data n1 and n2 respectively. Set values for each one. Write a program to interchange the values of both A and B. Use friend function.

20. Design a class SHAPE with dimensions d1 and d2 as member data and area() as member functions to find the area of a shape. Derive three classes RECT,TRIANG and CIRCL from the class SHAPE and override the function area() of base class to find the area of individual shape.Use virtual function.

21. Write a program to show returning current object, accessing member data of current object and returning values of object using this pointer.

22. Design a class employee with relevant emp details.Read the details of n emp from the keyboard and write it into a File named empdetails.At the end of writing every n emp details read them back from the same file and display into the screen.Use seperate functions to write and read into and out of the file.

23. Addition / Subtraction / Multiplication of complex numbers using classes.

24. Define a class to represent a bank account. Include the following members :

    Data Members

    1. Name of the depositor.
    2. Account number.
    3. Type of account.
    4. Balance amount in the account.

    Member Functions

    1. To assign initial values.
    2. To deposit an amount.
    3. To withdraw an amount after checking the balance.
    4. To display name and balance.

    Use appropriate main program.

25. Create two classes DM and DB which store the values of distances. DM stores distance in meters and centimeters and DB in feet and inches. Write a program that read values for the class objects. Include steps to add an object of DB with an object of DM. Use a friend function to carry out the addition and print the results in any unit.

26. Define a class string that could work as a user defined string type. Include constructors that will enable us to create an unintialized string and also to initialize an object with a string constant at the time of creation. Include a function to add two strings. Write a complete program to test your class to see that it does the following tasks :

    a. Create unitialized string objects.
    b. Create objects with string constants.
    c. Concatenate two string properly.
    d. Display the desire string objects.

27. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the object of FLOAT.

28. Assume that a bank maintain two types of accounts for customers, one called as saving account and the other as current account. The saving account provides

compound interest and withdrawal facilities, but no check book facility. The current account provides check book facility but no interest. Current account holders should maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class ACCOUNT that stores customer name, account number and type of account. From this derive the classes CURR_ACCT and SAVE_ACCT to make them more specific to their requirements. Include necessary member functions inorder to achieve the following tasks :

a. Accept deposit from a customer and update the balance.
b. Display the balance.
c. Compute and deposit interest.
d. Permit withdrawal and update balance.
e. Check for the minimum balance, impose penalty if necessary and update the balance.

Note : Do not use constructors. Use member functions to initialize the class members.

29. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called TRIANGLE and RECTANGLE from the base SHAPE. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived class to suite the requirements.

3C05CSC Data Base Management System

Contact Hours per Week : 3Theory
Credit : 2

Module I Introduction–Field,Record,Entity,Attribute,Relation,Domain,Tuple-Advantages of database systems- data models (Network model, Hierarchical Model, DBTG CODASYL model,Relational Model(E-R)) - system structure

Module 2 Database administrator- data base users, Constraints(Primary, Foreign, Candidate, Unique)-Relational Algebra (Union, Intersection, Difference, Product, Project, Selection)

Module 3 Normalization(First, Second, Third, Fourth, BCNF),SQL: Introduction To SQL-Tables-DDL,DML,DCL(In Detail),Data Types.

Module 4 SQL Functions(Different Types of Functions),Operators(Arithmetic, Relational, Logical), Sub Quires (in Detail),Clauses(Having, Group By).

Module 5 Joins(Different Types of Join Statements),View, Introduction to Sequence, Index and Triggers**.**

Textbook

1. Data Base Concept 3rd edition Abraham Silberschatz, Henery f Korth McGraw Hill
2. A Guide to the SQL Standard, C. J. Date and Hugh Darwen, 1997, Addison-Wesley

Reference:

1. An Introduction to Database Systems, C. J. Date, 1994, Addison-Wesley
2. Understanding the New SQL, Jim Melton and Alan R. Simon, 1993, Morgan Kaufmann.
3. Principles of Database & Knowledge Jeffrey D. Ullman, Computer Science Press, 1988.

3C06CSC Lab III (DBMS)

Contact Hours per Week	**:**	**2 Practical**
Credit	**:**	**1**

4C07CSC Visual Basic

Contact Hours per Week : 3 Theory
Credit : 2

Module 1 Visual Basic: What is Visual Basic , Structure of a VB Application,
Steps in developing Application, drawing the user interface and setting properties , setting properties of objects at design time and at runtime variables.

Module 2 VB data types , variable declaration, VB operators and functions, Branching statements – if then , goto, Looping statements, VB Tools, arrays, control arrays.

Module 3 Designing an application, using general sub procedures in applications, creating a code module, adding menus to an application, note editor, assigning icons to forms, creating VB Executable files, error types,

Module 4 Debugging VB programs, debugging strategies, sequential files, writing and reading text using sequential files. Random access files , writing and reading text using random access files , graphics methods, using colors, mouse events, timer tools and delays, animation techniques. Database structure and terminology, ADO data controls

Module 5 Connection strings, assigning tables, bound data tools, database management, customs controls, multiple form visual basic applications, VB multiple document interface (MDI), creating a help file, connection to the Database

Text Book

1. Visual Basic 6, G Cornell, Tata McGraw Hill

Reference

1. Programming windows by Charles PetZold
2. VC++ 6 from the ground up by John Paul Muller
3. VC++6 programming by Yaswant Kanitkar

4. mastering VC++ by Micheal J.Young BpB publications
5. MFC programming with VC++6 by David white,Kennscribner,Eugene olafsen.

3C08 CSC Lab IV (Visual Basic)

Contact Hours per Week : 2 Practical

Credit : 1

Sd/-
DR.G.Raju,
Chairman,BOS Computer Science(UG)