

KANNUR UNIVERSITY

B.Sc Programme

COMPLIMENTARY COURSES - COMPUTER SCIENCE

Semester	Code	Theory	Hours / Week	Credit
1	1C01CSC	Introduction to IT and C programming	2	2
1	1C02CSC	Lab – I (C Programming)	2	2
2	2C03CSC	Programming in C++	2	2
2	2C04CSC	Lab – II (C++ Programming)	2	2
3	3C05CSC	Data Base Management System And SQL	3	2
3	3C06CSC	Lab III (DBMs Lab)	2	2
4	4C07CSC	Visual Basic	3	2
4	4C08CSC	Lab IV (Visual Basic)	2	2

1C01CSC Introduction to IT and C Programming

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 :

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

Reference books:

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

1 C02CSC Lab I (C Programming)

List of Programs

Students have to record a minimum of 20 programs.

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. The straight line method of computing the yearly depreciation of the value of an item is given by

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

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

3. 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:

$$\begin{array}{r} 45 \\ X 37 \\ \hline 7 \times 45 \text{ is} 315 \\ 3 \times 45 \text{ is} 135 \\ \hline \text{Add them} 1665 \end{array}$$

4. Develop a program to find the number of and sum of all integers greater than 100 and less than 200 that are divisible by 7.
5. A set of two linear equations with two unknowns x_1 and x_2 is given below:

$$\begin{aligned} a x_1 + b x_2 &= m \\ c x_1 + d x_2 &= n \end{aligned}$$

The set has a unique solution

$$x_1 = \frac{md - bn}{ad - cb} \quad x_2 = \frac{na - mc}{ad - cb}$$

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 x_1 and x_2 . An appropriate message should be printed if $ad - cb = 0$.

6. 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.

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

Purchase amount Items	Discount	
	Mill cloth	Handloom
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.

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

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

conditional operator.

9. Develop a program using do—while loop to print the first m Fibonacci numbers.
10. 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 :
- $$E = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$$
14. Develop a program to evaluate the following function to 0.0001% accuracy.
- $$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$
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) , $i=1,2,\dots,n$. The straight line equation is $y = mx + c$. the values of m and c are given by:

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

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 three 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++

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 Structure, 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 (C++ Programming)

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 coordinates 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 separate 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 uninitialized 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 uninitialized string objects.
 - b. Create objects with string constants.
 - c. Concatenate two string properly.
 - d. Display the desired 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 in order 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 & SQL

Module 1

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

4C07CSC Visual Basic

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.