7	Basic Programmin in C++	SEC0200703	3	40-60

FYUGP, SEC syllabus, 2nd Semester

SEC0200703: Basic Programming in C++

Credit: 3 **Evaluation: 40-60**

Program Outcome (PO):

Students, who choose FYUGP SEC Computer related Programme will develop the ability to think critically, logically, analytically and to use and apply current technical concepts and practices in the core development of solutions in the form of Information Technology. The knowledge and skills gained with a FYUGP graduates for a broad range of jobs in Education sector, Research field, Government sector, Business sector and Industry.

Hands on sessions in Computer Lab using C++ Programming languages and tools will enable students to deal with real life problems which will lead to better understanding of the topics and will also widen the horizon of students' self-experience.

Program Specific Outcomes (PSOs) :

Completion of FYUGP SEC Programme shall enable a student:–

(1) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
(2) Apply the knowledge gained in Computer related SEC courses to a broad range of advanced topics in Computer Science & IT, to learn and develop sophisticated technical products independently.

(3) To design, implement, and evaluate computer-based system, process, component, or program to meet desired needs by critical understanding, analysis and synthesis.
(4) Identify applications of Computer Science in other related fields in the real world to enhance the career prospects.

Course Outcome (CO):

On successful completion of this subject the students have the Basic fundamental concepts of the Computer Programming ability in C++ Language.

This paper helps students to inculcate knowledge on the basic concepts of C++ programming includes arrays, structures, function, strings, and files.

- Understand the basic terminology used in computer programming.
- Write, compile and debug programs in C++ language.
- Create programs involving decision structures & unions, loops, strings and functions.
- Design programs involving structures and pointers.

Course Designer:

Dr. Masih Saikia, HoD, CSC Department,

Pragjyotish College, Guwahati-781009.Ph: 9613974770, 8638283356

Detail Syllabus:

Unit-1: Data Types, Variables, Operators and Statements

C++ Character Set, Concept of Data-types, Identifiers and Keywords, Variables (declaration and initialization), Constants: (String, Numeric, and Character Constants), Operators: (Arithmetic, Assignment, Increment and Decrement, Comparison/Relational, Logical, Bitwise, Special Operators), Type Conversion.

Unit-2 : Writing a Program in C++

Declaration of Variables, Statements, Simple C++ Programs, Features of iostream.h, Keyboard and Screen I/O, Using I/O Operators: (output operator "<<", Input operator ">>") Cascading of I/O operators.

Unit-3 : Control Statements / Flow of Control

Conditional Expressions/Selection Statements: (if, if-else, switch-case-default statements), Iteration/Loop Statements: (for, while, do-while loops, nested loop), Breaking Control/Jump Statements: (break, continue, goto statement, exit( ) function)

Unit-4 : Functions and Program Structures

Defining of a Function, return statement, Types of Functions, Accessing a Function : Actual and Formal Parameters/Arguments, Local and Global Variables. Default Arguments, Multi-function Program, Recursive Function. Common C++ Preprocessor directives, Header Files: (stdio.h, iostream.h, ctype.h, strings.h, maths.h, stdlib.h)

Unit-5 : Arrays

Array Notation, Array Declaration, Array Initialization, Processing with Array, Array and Functions, Multi-dimensional Arrays.

Unit-6 : Character handling in C++ and Strings

Character Array, Declaration of String variables, Reading strings, String handling [without using library functions] : (1) finding the length of the string, (2) String concatenation, (3) String reverse, (4) String copy, (5) Combining/Joining strings together, (6) String comparison, (7) Extraction of a string from another string. Using of the String handling C++ library functions: strcpy(), strcat(), strlen(), strcmp().

Unit-7 : Structure and Union

Declaration of Structure, Initialization of Structure, Referencing Structure Elements, Arrays of Structures, Arrays within a Structure, Unions, typedef keyword.

Unit-8 : Data File Handling in C++

Input/Output operations on files, Opening and Closing of Files, Writing and Reading Characters using put() and get() functions, Binary input/output file operation using read() and write().

Unit-9:

Concept of objects and classes. structure versus class, member functions, simple example of classes with member function.

Practical / Lab work to be performed

(N.B: Student has to perform **any twenty** of the following experiments)

1. Write a C++ program to find the maximum, minimum and sum of n given numbers without using array.
2. Write a C++ program to print Multiplication Table of a given number.
3. Write a C++ program to input number of Week's day (1-7) and translate to its equivalent name of the day of the week (e.g., 1 to Sunday, 2 to Monday, … 7 to Saturday).
4. Write a C++ program to perform Arithmetic Calculator using switch-case statement. This Program inputs two operands and an operator and then displays the calculated result.
5. Write a C++ program for Temperature conversion that gives the user the option of converting Fahrenheit to Celsius or Celsius to Fahrenheit and depending upon the user's choice carries out the conversion.

6. Write a C++ program to calculate area of a Circle, a Rectangle or a Triangle depending upon the user's choice.

7. Write a C++ program to checks whether a given number is Palindrome or not.

8. Write a program in C++ to calculate **Factorial** of a number : (i) using *recursion*, (ii) using *iteration*

9. Write a program in C++ to display **Fibonacci** series (i) using *recursion*, (ii) using *iteration*

10. Write a function in a C++ program to find the **L.C.M.** and **G.C.D.** of given two numbers.

11. Write a program in C++ to find the sum of the digits of given number.

12. Write a C++ program to check whether a given number is **Armstrong** number or not.

    (Hint : Armstrong number is one which is equal to the sum of cube of the individual digits.

    For example : 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153)

13. Write a C++ program to check whether a given number is a **Perfect** number or not.

    (Hint : Perfect number is one which is equal to the sum of its factors.

    For example : Factors of 28 : 1, 2, 4, 7, 14. Sum of the factors of 28 = 1+2+4+7+14 = 28)

14. Write a C++ program to find the Roots of a Quadratic Equation. Include all possibilities.

15. Write a C++ program to check whether a given number is prime or not using exit() function.

16. Write a C++ program to print the first 'n' prime numbers.

17. Write a C++ program to sort the elements of an array (one-dimensional) in ascending order [use 'Bubble sort' method].

18. Write a C++ program to merge two arrays after eliminating duplicate elements.

19. Write a C++ program to transpose a matrix.

20. Write a C++ program to multiply two matrices of order L×M and M×N.

21. Write a C++ program to find row sum and column sum of an N×M matrix.

22. To demonstrate the use of three-dimensional array, write a C++ program to input semester wise marks of different papers for different test exams. (Say you have 2 semesters; 4 subjects per semester; and 3 tests per subject). Now display (i) paper-wise total marks, (ii) Semester-wise total marks and (iii) Grand total of the Marks awarded to the student.

23. Write a C++ program to convert number into words. (say when you input 3926, you will get the output as THREE THOUSAND NINE HUNDR AND TWENTY SIX ONLY)

24. Write a function in a C++ program to find the maximum number in an array.

25. Write a program in C++ to demonstrate the use of some of the frequently used mathematical functions (say sqrt(), pow(), exp(), sin(), cos(), tan(), log(), abs(), fabs(), floor(), ceil()]

26. Write a program in C++ to convert a Binary number to a Decimal:

27. Write a program in C++ to convert Decimal number to any base.

28. To demonstrate the use of two-dimensional array of strings, write a C++ program to input a few strings (say input the names of 5 different Cities), sort them, and display the strings in alphabetical order.

29. Write a program in C++ to find and replace a string.

30. Write a program in C++ to check whether a given string is palindrome or not.

31. Write a C++ program to find number of vowels in a given line of text.

32. Write a program in C++ to count the number of letters and words in a given string.

33. Write a program in C++ to compute the **sine** series :

    sin(x) = x – (x^3)/3! + (x^5)/5! – .. (x^n)/n!

34. Write a C++ program to count the number of spaces in a string.

35. To Demonstrate the **sorting** of Arrays of Structures, Create a structure ***School*** containing fields for *Name*, *RollNo*, *Gender*, *Height*, and *Weight*. Enter at least 4 **School** data records, sort the records alphabetically on the basis of the names in ascending order; display them records neatly.

36. To Demonstrate the use of Arrays of Structures, Create a structure ***Employee*** containing fields for *EmpName*, *EmpNo*, *BasicPay*. Enter at least 5 employees' data records and display them neatly.

37. Write a C++ program (Telephone Directory simulation) that can sort a list of names and telephone number alphabetically. Names to be treated as a unit. (define them inside a structure). Persons are sorted alphabetically by their last names. Persons with the same last name are sorted by their first names.

38. Write a C++ program to count number of vowels present in a text file.

39. Write a C++ program to Write and Read a structure using write() and read() functions using a binary file.

40. Define a class named "rectangle" to represent rectangles in a 2- dimensional plane. For each rectangle, its length and breadth are to be stored. Write 2 different member functions, one to computer the area of a rectangle and its perimeter and the other to check if a rectangle is a square.

Reference Books :

1. Herbtz Schildt, "C++: The Complete Reference", 4th Edition, McGraw Hill, 2017.

2. Bjarne Stroustrup, "The C++ Programming Language", 4th Edition, Pearson Education, 2022.

3. E Balaguruswamy, "Object Oriented Programming with C++", 8th Edition, Tata McGraw-Hill Education, 2020.

5. D. Ravichandran, "Programming with C++", Tata McGraw-Hill Publishing Co. Ltd, 3rd Edition, 2017.