

Syllabus for Computer Science

Proposed scheme for **B.Sc.** Programme under **Choice Based Credit System**

| Code | Course Title | Course Type | HpW | Credits |
|-----------------------|----------------------------------|-------------|---------|----------|
| SEMESTER – I | | | | |
| BS106 | Programming in C | DSC-3A | 4T+2P=6 | 4 + 1 =5 |
| SEMESTER – II | | | | |
| BS206 | Programming in C++ | DSC-3B | 4T+2P=6 | 4 + 1 =5 |
| SEMESTER – III | | | | |
| BS301 | A: SciLab – 1 | SEC-1 | 2 | 2 |
| | B: Boolean Algebra | | | |
| BS306 | Data Structures | DSC-3C | 4T+2P=6 | 4 + 1 =5 |
| SEMESTER – IV | | | | |
| BS401 | C: SciLab – 2 | SEC-2 | 2 | 2 |
| | D: Digital Logic | | | |
| BS406 | Database Management Systems | DSC-3D | 4T+2P=6 | 4 + 1 =5 |
| SEMESTER – V | | | | |
| BS501 | Information Technologies –1 | GE-1 | 2 | 2 |
| BS502 | E: Python – 1 | SEC-3 | 2 | 2 |
| | F: Computer Organization | | | |
| BS505 | Programming in Java | DSC-3E | 3T+2P=5 | 3 + 1 =4 |
| BS506 | Elective–A: Operating Systems | DSE-1E | 3T+2P=5 | 3 + 1 =4 |
| | Elective–B: Software Engineering | DSE-2E | 3T+2P=5 | 3 + 1 =4 |
| SEMESTER – VI | | | | |
| BS601 | Information Technologies –2 | GE-2 | 2 | 2 |
| BS602 | G: Python – 2 | SEC-4 | 2 | 2 |
| | H: Numerical Computing | | | |
| BS605 | Computer Networks | DSC-3F | 3T+2P=5 | 3 + 1 =4 |
| BS606 | Elective–A: PHP with MySQL | DSE-1F | 3T+2P=5 | 3 + 1 =4 |
| | Elective–B: Web Technologies | DSE-2F | 3T+2P=5 | 3 + 1 =4 |

DSC-3A

Programming in C

BS106

Theory: 4 credits and Practical: 1 credit
Theory: 4 Hours/Week and Practical: 2 Hours/Week

Unit – I

Computer Fundamentals: Introduction of Computers, Classification of Computers, Anatomy of a Computer, Memory Hierarchy, Introduction to OS, Operational Overview of a CPU.

Program Fundamentals: Generation and Classification of Programming Languages, Compiling, Interpreting, Loading, Linking of a Program, Developing Program, Software Development.

Algorithms: Definitions, Different Ways of Stating Algorithms (Step-form, Pseudo-code, Flowchart), Strategy for Designing Algorithms, Structured Programming Concept.

Basics of C: Overview of C, Developing Programs in C, Parts of Simple C Program, Structure of a C Program, Comments, Program Statements, C Tokens, Keywords, Identifiers, Data Types, Variables, Constants, Operators and Expressions, Expression Evaluation–precedence and associativity, Type Conversions.

Unit – II

Input-Output: Non-formatted and Formatted Input and Output Functions, Escape Sequences,

Control Statements: Selection Statements – if, if-else, nested if, nested if-else, comma operator, conditional operator, switch; Iterative Statements–while, for, do-while; Special Control Statement–goto, break, continue, return, exit.

Arrays and Strings: One-dimensional Arrays, Character Arrays, Functions from ctype.h, string.h, Multidimensional Arrays.

Unit – III

Functions: Concept of Function, Using Functions, Call-by-Value Vs Call-by-reference, Passing Arrays to Functions, Scope of Variables, Storage Classes, Inline Functions, and Recursion.

Pointers: Introduction, Address of Operator (&), Pointer, Uses of Pointers, Arrays and Pointers, Pointers and Strings, Pointers to Pointers, Array of Pointers, Pointer to Array, Dynamic Memory Allocation.

Unit – IV

User-defined Data Types: Declaring a Structure (Union) and its members, Initialization Structure (Union), Accessing members of a Structure (Union), Array of Structures (Union), Structures Vs Unions, Enumeration Types.

Files: Introduction, Using Files in C, Working with Text Files, Working with Binary Files, Files of Records, Random Access to Files of Records, Other File Management Functions.

Text Pradip Dey, Manas Ghosh, *Computer Fundamentals and Programming in C* (2e)

References

1. Ivor Horton, *Beginning C*
2. Herbert Schildt, *The Complete Reference C*
3. Paul Deitel, Harvey Deitel, *C How To Program*
4. Byron S. Gottfried, *Theory and Problems of Programming with C*
5. Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*
6. B. A. Forouzan, R. F. Gilberg, *A Structured Programming Approach Using C*

BS106

C Lab

Practical: 2 Hours/Week

Credit: 1

1. Write a program to find the largest two (three) numbers using if and conditional operator.
2. Write a program to print the reverse of a given number.
3. Write a program to print the prime number from 2 to n where n is given by user.
4. Write a program to find the roots of a quadratic equation using switch statement.
5. Write a program to print a triangle of stars as follows (take number of lines from user):

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

6. Write a program to find largest and smallest elements in a given list of numbers.
7. Write a program to find the product of two matrices..
8. Write a program to find the GCD of two numbers using iteration and recursion.
9. Write a program to illustrate use of storage classes.
10. Write a program to demonstrate the call by value and the call by reference concepts.
11. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program to illustrate use of data type enum.
13. Write a program to demonstrate use of string functions string.h header file.
14. Write a program that opens a file and counts the number of characters in a file.
15. Write a program to create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
16. Write a program that opens an existing text file and copies it to a new text file with all lowercase letters changed to capital letters and all other characters unchanged.

Note:

1. Write the Pseudo Code and draw Flow Chart for the above programs.
2. Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows 10.

DSC-3B

Programming in C++

BS206

Theory: 4 credits and Practical: 1 credit
Theory: 4 Hours/Week and Practical: 2 Hours/Week

Unit – I

Introduction to C++: Applications, Example Programs, Tokens, Data Types, Operators, Expressions, Control Structures, Arrays, Strings, Pointers, Searching and Sorting Arrays.

Functions: Introduction, Prototype, Passing Data by Value, Reference Variables, Using Reference Variables as Parameters, Inline Functions, Default Arguments, Overloading Functions, Passing Arrays to Functions.

Object Oriented Programming: Procedural and Object-Oriented Programming, Terminology, Benefits, OOP Languages, and OOP Applications.

Unit – II

Classes: Introduction, Defining an Instance of a Class, Why Have Private Members? Separating Class Specification from Implementation, Inline Member Functions, Constructors, Passing Arguments to Constructors, Destructors, Overloading Constructors, Private Member Functions, Arrays of Objects, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading, Object Conversion, Aggregation.

Unit – III

Inheritance: Introduction, Protected Members and Class Access, Base Class Access Specification, Constructors and Destructors in Base and Derived Classes, Redefining Base Class Functions, Class Hierarchies, Polymorphism and Virtual Member Functions, Abstract Base Classes and Pure Virtual Functions, Multiple Inheritance.

C++ Streams: Stream Classes, Unformatted I/O Operations, Formatted I/O Operations.

Unit – IV

Exceptions: Introduction, Throwing an Exception, Handling an Exception, Object-Oriented Exception Handling with Classes, Multiple Exceptions, Extracting Data from the Exception Class, Re-throwing an Exception, Handling the `bad_alloc` Exception.

Templates: Function Templates–Introduction, Function Templates with Multiple Type, Overloading with Function Templates, Class Templates – Introduction, Defining Objects of the Class Template, Class Templates and Inheritance, Introduction to the STL.

Text Tony Gaddis, *Starting out with C++: from control structures through objects (7e)*

References

1. B. Lippman, *C++ Primer*
2. Bruce Eckel, *Thinking in C++*
3. K.R. Venugopal, *Mastering C++*
4. Herbert Schildt, *C++: The Complete Reference*
5. Bjarne Stroustrup, *The C++ Programming Language*
6. Sourav Sahay, *Object Oriented Programming with C++*

BS206

C++ Lab

Practical: 2 Hours/Week

Credit: 1

1. Write a program to.
 - a. Print the sum of digits of a given number.
 - b. Check whether the given number is Armstrong or not
 - c. Print the prime number from 2 to n where n is natural number given.
2. Write a program to find largest and smallest elements in a given list of numbers and sort the given list.
3. Write a menu driven program that can perform the following functions on strings. (Use overloaded operators where possible).
 - a. Compare two strings for equality (== operator)
 - b. Check whether first string is smaller than the second (<= operator)
 - c. Copy the string to another.
 - d. Extract a character from the string (overload [])
 - e. Reverse the string.
 - f. Concatenate two strings (+ operator)
4. Write a program using friend functions and inline functions.
5. Write a program to find area of a rectangle, circle, and square using constructors.
6. Write a program to implement copy constructor.
7. Write a program to demonstrate single inheritance and multiple inheritances.
8. Write a program to demonstrate hierarchical inheritance and multipath inheritance (using virtual functions)
9. Write a program to demonstrate static polymorphism using method overloading.
10. Write a program to demonstrate dynamic polymorphism using method overriding and dynamic method dispatch.
11. Write a program to demonstrate the function templates and class templates.
12. Write a program to menu driven program for accepting two numbers and perform calculator operations addition, subtraction, multiplication, division and remainder using function template.
13. Write a program to demonstrate exception handling.
14. Write a program to demonstrate various input-output manipulations.
15. Write a program to implement stack abstract data type.
16. Write a program to demonstrate array of objects.

Note: Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows 10.

MOOCs (Massive Online Open Courses) Free Resources

E-Learning:

- NPTEL :nptel.ac.in [Core Subjects Certification]
- C++ INSTITUTE :cppinstitute.org [C++ Certification]
- ORACLEEDUCATION :education.oracle.com [Java, DBMS Certification]
- BIG DATA UNIVERSITY :bigdatauniversity.com [Big Data Certification]
- COURSERA :coursera.org [Core Subjects Certification]
- CODEACADEMY :codecademy.com [Coding Certification]
- KHANACADEMY :khanacademy.org [Core Subjects Certification]
- PIXAR IN A BOX :khanacademy.org/partner-content/pixar
- VIDEOLECTURES :videlectures.net
- YOUTUBEEDU :plus.google.com/+YouTubeEDU/posts
- DISNEY RESEARCH :disneyresearch.com
- ALISON :alison.com [Core Subjects Certification]
- INTERNET ARCHIVE :archive.org
- C++PROGRAMMING :cprogramming.com [Learning C and C++]

Freeware:

- SCILAB : scilab.org [MatLab Equivalent]
- GEOGEBRA :geogebra.org [Software for Class Room Teaching]

Search Engine:

- WOLFRAM ALPHA :wolframalpha.com [Computing Engine]
- CITESEER :citeseerx.ist.psu.edu [Searching Research Articles]
- DOAJ :doaj.org [Open Access to Journals]