

Syllabus for Computer Applications

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: Python – 1			
BS306	Relational Database Management Systems	DSC-3C	4T+2P=6	4 + 1 =5
SEMESTER – IV				
BS401	C: SciLab – 2	SEC-2	2	2
	D: Python – 2			
BS406	Computer Networks	DSC-3D	4T+2P=6	4 + 1 =5
SEMESTER – V				
BS501	Information Technologies – 1	GE-1	2	2
BS502	E: R Basics – 1	SEC-3	2	2
	F: Ruby			
BS505	Multimedia Systems	DSC-3E	3T+2P=5	3 + 1 =4
BS506	Elective–A: Web Technologies	DSE-1E	3T+2P=5	3 + 1 =4
	Elective–B: Visual Programming	DSE-2E	3T+2P=5	3 + 1 =4
SEMESTER – VI				
BS601	Information Technologies – 2	GE-2	2	2
BS602	G: R Basics – 2	SEC-4	2	2
	H: Ruby on Rails			
BS605	Mobile Applications	DSC-3F	3T+2P=5	3 + 1 =4
BS606	Elective–A: PHP Programming	DSE-1F	3T+2P=5	3 + 1 =4
	Elective–B: Information Security and Cyber Laws	DSE-2F	3T+2P=5	3 + 1 =4

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 and Two Dimensional Arrays, Character Arrays, Functions from ctype.h, string.h.

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, 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), Structures Vs Unions, Enumeration Types.

Files: Introduction, Using Files, Working with Text Files and Binary Files, 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 numbers using if and conditional operator.
2. Write a program to calculate arithmetic operations of two numbers using switch.
3. Write a program to print the reverse of a given number.
4. Write a program to print whether the given number is a prime or not.
5. Write a program to find largest and smallest elements in a given list of numbers.
6. Write a program to find the sum of two matrices.
7. Write a program to find the product of two matrices.
8. Write a program to print the reverse of a given string.
9. Write a program to find the factorial of a positive integer using iteration and recursion.
10. Write a program to find the GCD of two positive integers using iteration and recursion.
11. Write a program to demonstrate the call by value and the call by reference concepts.
12. Write a program to illustrate the use of Enumeration data type.
13. Write a program to illustrate the use of structure concept.
14. Write a program to illustrate the use of union concept.
15. Write a program to write content into a file and display contents of a file
16. Write a program to copy content of one file into another file and display the content of new file.

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.

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, Variables, Operators, Expressions, Control Structures, Arrays, Strings, Pointers.

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

Unit – II

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

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, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading.

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, 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.

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.

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 print the sum of digits of a given number
2. Write a program to check whether the given number is Armstrong or not
3. Write a program to check whether the given string is Palindrome or not
4. Write a program to read the student name, roll no, marks and display the same using class and object.
5. Write a program to find area of a rectangle, circle, and square using class and object.
6. Write a program to implement inline function inside and outside of a class for
 - a. Finding the area of a square
 - b. Finding the area of a cube
7. Write a program to implement friend function and friend class
8. Write a program to implement constructor and destructor with in a class.
9. Write a program to demonstrate hierarchical inheritance.
10. Write a program to demonstrate multiple inheritances.
11. Write a program to demonstrate the constructor overloading.
12. Write a program to demonstrate static polymorphism.
13. Write a program to demonstrate dynamic polymorphism.
14. Write a program to implement polymorphism using pure virtual functions.
15. Write a program to demonstrate the function templates and class templates.
16. Write a program to demonstrate exception handling using try, catch, and finally.

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

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]