# COMPUTER SCIENCE

**Course Structure**

| Units | Topics | Marks |
|---|---|---|
| 1 | Computer Fundamentals | 10 |
| 2 | Programming Methodology | 12 |
| 3 | Introduction to Python/C++ | 18 |
| 4 | Programming with Python/C++ | 30 |
| **Total** | | **70** |

## Unit 1: Computer Fundamentals

### Chapter 1: Classification of Computers

- ➢ Basics of computer and its operation
- ➢ Functional components and their interconnections
- ➢ Concept of booting

### Chapter 2: Software Concepts:

- ➢ Types of Software
  - System software
  - Utility software
  - Application software

### Chapter 3: System Software

- ➢ Operating system
- ➢ Complier
- ➢ Interpreter and assembler

**Chapter 4: Operating System**

- ➤ Need for operating system
- ➤ Functions of operating system:
  - Processor management
  - Memory management
  - File management
  - Device management
- ➤ Types of operating system-interactive (GUI based)
- ➤ Time sharing
- ➤ Real time and distributed
- ➤ Commonly used operating system:
  - UNIX
  - LINUX
  - Windows
  - Solaris
  - BOSS (Bharat Operating System Solutions)
- ➤ Mobile OS
  - Android
  - Symbian

**Chapter 5: Utility Software**

- ➤ Anti-Virus
- ➤ File Management tools
- ➤ Compression tools
- ➤ Disk Management tools:
  - Disk Cleanup
  - Disk Defragmenter Backup

**Chapter 6: Open Source Concepts**

- ➤ Open source software
- ➤ Freeware, shareware
- ➤ Proprietary software

## Chapter 7: Application Software

- ➢ Office tools:
  - Word processor
  - Presentation tool
  - Spreadsheet package
  - Database management system
- ➢ Domain specific tools:
  - School management system
  - Inventory management system
  - Payroll system
  - Financial accounting
  - Hotel management
  - Reservation system
  - Weather forecasting system

## Chapter 8: Number System

- ➢ Binary
- ➢ Octal
- ➢ Decimal
- ➢ Hexadecimal
- ➢ Conversion between two different number systems

## Chapter 9: Internal Storage encoding of Characters

- ➢ ASCII
- ➢ ISCII (Indian Scripts Standard Code for Information Interchange)
- ➢ UNICODE (for multilingual computing)

## Chapter 10: Microprocessor

- ➢ Basic concepts
- ➢ Clock speed (MHz, GHz):
  - 16 bit
  - 32 bit

- 64 bit
- 128 bit processors

➢ Types:

- CISC Processors (Complex Instruction Set Computing)
- RISC Processors (Reduced Instruction Set Computing)
- EPIC (Explicitly Parallel Instruction Computing)

## Chapter 11: Memory Concepts

➢ Units:

- Byte
- Kilo Byte
- Mega Byte
- Giga Byte
- Tera Byte
- Peta Byte
- Exa Byte
- Zetta Byte
- Yotta Byte

## Chapter 12: Primary Memory

➢ Cache
➢ RAM
➢ ROM

## Chapter 13: Secondary Memory

➢ Fixed and Removable storage

- Hard Disk Drive
- CD/DVD Drive
- Pen Drive
- Blue Ray Disk

**Chapter 14: Input Output Ports/ Connections**

- ➢ Serial
- ➢ Parallel and Universal Serial Bus
- ➢ PS-2 port
- ➢ Infrared port
- ➢ Bluetooth
- ➢ Firewire

# Unit 2: Programming Methodology

**Chapter 15: General Concepts**

- ➢ Clarity and simplicity of expressions
- ➢ Use of proper names for identifiers
- ➢ Comments
- ➢ Indentation
- ➢ Documentation
- ➢ Program maintenance
- ➢ Running and debugging programs
- ➢ Syntax errors
- ➢ Run-time errors
- ➢ Logical errors

**Chapter 16: Problem solving methodologies**

- ➢ Understanding of the problem
- ➢ Solution for the problem
- ➢ Breaking down solution into simple steps (modular approach)
- ➢ Identification of arithmetic and logical operations required for solution
- ➢ Control structure- conditional control and looping (finite and infinite)

**Chapter 17: Problem Solving**

- ➢ Introduction to algorithms/flowcharts

## Unit 3: Introduction to Python

### Chapter 18: Getting Started

- ➢ Introduction to Python:
    - • An integrated high level language
    - • Interactive mode and script mode
- ➢ Data types:
    - • Number (Integer - boolean, decimal, octal, hexadecimal; Floating point; Complex), none, Sequence (String, Tuples, List)
    - • Sets
    - • Mapping

### Chapter 19: Mutable and Immutable Variables

### Chapter 20: Variables, Expressions and Statements

- ➢ Values, Variables and keywords
- ➢ Operators and Operands in Python: (Arithmetic, relational and logical operators)
- ➢ operator precedence
- ➢ Expressions and Statements (Assignment statement)
- ➢ Taking input (using raw_input() and input()) and displaying output (print statement)
- ➢ Putting Comments

### Chapter 21: Functions

- ➢ Importing Modules (entire module or selected objects)
- ➢ Invoking built in functions
- ➢ Functions from math module (for example, ceil, floor, fabs, exp, log, log10, pow, sqrt, cos, sin, tan, degrees, radians)
- ➢ Using random() and randint() functions of random module to generate random numbers
- ➢ Composition

### Chapter 22: Defining functions

➤ Invoking functions

➤ Passing parameters (default parameter values, keyword arguments)

➤ Scope of variables

➤ Void functions and functions returning values

➤ Flow of execution

### Chapter 23: Conditional constructs and looping

➤ If else statement while

➤ For (range function)

➤ Break

➤ Continue

➤ Else

➤ Pass

➤ Nested if

➤ Nested loops

➤ Use of compound expression in conditional and looping construct


## Unit 3: Introduction to C++

### Chapter 24: Getting Started

➤ C++ character set

➤ C++ Tokens (Identifiers, Keywords, Constants, Operators,)

➤ Structure of a C++ Program (include files, main function)

➤ Header files – iostream.h, iomanip.h, cout, cin

➤ Use of I/O operators (<<and>>)

➤ Use of endl and setw ( )

➤ Cascading of I/O operators

➤ Compilation

➤ Error Messages

➤ Use of editor

- ➤ Basic commands of editor
- ➤ Compilation
- ➤ Linking
- ➤ Execution

## Chapter 25: Data Types, Variables and Constants

- ➤ Concept of Data types
- ➤ Built-in Data types: char, int, float and double
- ➤ Constants: Integer Constants, Character constants - \n, \t, \b), Floating Point Constants, String Constants
- ➤ Access modifier
- ➤ Variables of built-in-datatypes
- ➤ Declaration/Initialization of variables
- ➤ Assignment statement
- ➤ Type modifier: signed, unsigned, long

## Chapter 26: Operator and Expressions: Operators

- ➤ Arithmetic operators (-,+,*,/,%)
- ➤ Assignment operator(=)
- ➤ C++ shorthands (+=,- =,*=,/=,%=)
- ➤ Unary operator (-)
- ➤ Increment(++) and Decrement (--) Operators
- ➤ Relation operator (>,>=,<=,=,!=)
- ➤ Logical operators (!,&&,II)
- ➤ Conditional operator
- ➤ Precedence of Operators
- ➤ Automatic type conversion in expressions
- ➤ Type casting

**Unit 4: Programming with Python**

**Chapter 27: Strings**

- ➢ Creating
- ➢ Initialising and accessing the elements
- ➢ String operators:
  - • +, *, in, not in, range slice [n:m]
  - • Comparing strings using relational operators
  - • String functions & methods: len, capitalize, find, isalnum, isalpha, isdigit, lower, islower, isupper, upper, lstrip, rstrip, isspace, istitile, partition, replace, join, split, count, decode, encode, swapcase, String constants, Regular Expressions and Pattern Matching

**Chapter 28: Lists**

- ➢ Concept of mutable lists
- ➢ Creating
- ➢ Initializing and accessing the elements
- ➢ Traversing
- ➢ Appending
- ➢ Updating and deleting elements
- ➢ Composition
- ➢ Lists as arguments

**Chapter 29: List operations**

- ➢ Joining
- ➢ Slicing
- ➢ +
- ➢ *
- ➢ in
- ➢ not in

## Chapter 30: List functions and methods:

- len( )
- insert( )
- append( )
- extend( )
- sort( )
- remove( )
- reverse( )
- pop( )
- list( )
- count( )
- extend( )
- index( )
- cmp( )
- max( )
- min( )

## Chapter 31: Dictionaries

- Concept of key-value pair
- Creating, initialising and accessing the elements in a dictionary
- Traversing
- Appending
- Updating
- Deleting elements

## Chapter 32: Dictionary functions and methods

- cmp( )
- len( )
- clear( )
- get( )
- has_key( )
- items( )

- ➢ key( )
- ➢ update( )
- ➢ values( )
- ➢ pop( )
- ➢ fromkeys( )
- ➢ dict( )

## Chapter 33: Tuples

- ➢ Immutable concept
- ➢ Creating
- ➢ Initialising and accessing elements in a tuple
- ➢ Tuple assignment
- ➢ Tuple slices
- ➢ Tuple indexing

## Chapter 34: Tuple Functions

- ➢ cmp()
- ➢ len()
- ➢ max()
- ➢ min()
- ➢ tuple()
- ➢ index()
- ➢ count()
- ➢ sum()
- ➢ any()
- ➢ all()
- ➢ sorted()
- ➢ reversed()

# Unit 4: Programming In C++

## Chapter 35: Flow of control

➢ Conditional statements:

- if else
- Nested if
- switch..case..default
- Nestedswitch..case
- break statement (to be used in switch..case only)
- Loops: while, do - while, for and Nested loops

## Chapter 36: Inbuilt Functions

➢ Standard input/output functions - stdio.h: gets ( ), puts ( )

➢ Character Functions - Ctype.h: isalnum ( ), isalpha ( ),isdigit ( ), islower ( ), isupper ( ), tolower ( ), toupper ( )

➢ String Function - string.h: strcpy ( ), strcat ( ), strlen ( ), strcmp ( ), strcmpi ( ), strev ( ),strlen ( ), strupur ( ), strlwr ( )

➢ Mathematical Functions - math.h: fabs ( ), pow ( ), sgrt ( ), sin ( ), cos ( ), abs ( )

➢ Other Functions - stdlib.h: randomize ( ), random ( )

## Chapter 37: Introduction to user-defined function and its requirements

➢ Defining a function

➢ Function prototype

➢ Invoking/calling a function

➢ Passing arguments to function

➢ Specifying argument data types

➢ Default argument

➢ Constant argument

➢ Call by value

➢ Call by reference

➢ Returning values from a function

- Scope rules
- Local and global variables
- Relating to Parameters and return type concepts in built-in functions

## Chapter 38: Structured Data Type

- Arrays:
  - Introduction
  - Advantages
- One Dimensional Array:
  - Declaration/initialization of One-dimensional array
  - Inputting array elements
  - Accessing array elements
  - manipulation of array elements (sum of elements, product of elements, average of elements linear search, finding maximum/minimum value)
  - Declaration / Initialization of a String
  - String manipulations (counting vowels/ consonants/ digits/ special characters, case conversion, reversing a string, reversing each word of a string)
- Two-dimensional Array:
  - Declaration/initialization of a two-dimensional array
  - Inputting array elements accessing array elements
  - Manipulation of array elements (sum of row element, column elements, diagonal elements, finding maximum / minimum values)
- User-defined Data Types:
  - Introduction to user defined data types
- Structure:
  - Defining a Structure
  - Declaring structure variables
  - Accessing structure elements
  - Passing structure to functions as value and reference
  - Function returning structure
  - Array of structure

➢ Defining a symbol name using typed ef keyword and defining a macro using #define preprocessor directive