PROGRAMMING COURSE SYLLABUS

C Language Syllabus

Fundamentals in C

- Program
- Programming
- Programming Languages
- Types of software
- Introduction to C
- History of C
- Features of C
- Applications of C
- Character set, ASCII Table
- Tokens
- Keywords
- Identifiers & Naming Rules
- constants
- Data Types
- Type Qualifiers
- How does the data stored in Computers Memory
- Variables
- Variable Declaration
- Variable Assignment
- Variable Initialization
- Comments
- Defining Constants
- MCQs

Operators and Expressions

- Arithmetic operators
- Arithmetic expressions
- Evaluation of expressions
- Relational operators
- Logical operators

- Assignment operators
- Increment & decrement operators
- Conditional operator
- Bitwise operators
- Type casting
- Sizeof operator
- Comma operator
- Operators Precedence and Associativity
- Expressions
- Evaluation of Expressions
- MCQs

Input-Output Functions

- Input-Output Library Functions
- Non-formatted Input and Output
- Character oriented Library functions
- Compiler, Linker and Loader
- Program execution phases
- Formatted Library Functions
- Mathematical Library Functions
- Structure of a C Program
- IDE
- Basic programs
- MCQs

Control Statements

- Conditional Control Statements
  - if
  - if-else
  - nested if-else
  - if-else-if ladder
- Multiple Branching Control Structure
  - switch-case
- Loop Control statements
  - while
  - do-while

    - for
- Nested Loops
- Jump Control structures
- break
- continue
- goto
- return
- Programs
- MCQs

Arrays

- Arrays
- One dimensional arrays
- Declaration of 1D arrays
- Initialization of 1D arrays
- Accessing element of 1D arrays
- Reading and displaying elements
- Programs on 1D Arrays
- Two dimensional arrays
- Declaration of 2D arrays
- Initialization of 2D arrays
- Accessing element of 2D arrays
- Reading and displaying elements
- Programs on 2D Arrays
- Three dimensional arrays
- MCQs

Strings

- String Concept
- Introduction to String in C
- Storing Strings
- The string Delimiter
- String Literals (String Constants)
- Strings and Characters
- Declaring Strings
- Initializing Strings

- Strings and the Assignment Operator
- String Input Functions / Reading Strings
- String Output Functions / Writing Strings
- String Input-Output using fscanf() and fprintf() Functions
- Single Character Library Functions / Character Manipulation in the String
- String Manipulation Library Functions
- Programs Using Character Arrays
- Array of Strings (2D Character Arrays)
- Programs Using Array of Strings
- MCQs

Pointers

- Understanding Memory Addresses
- Pointer Operators
- Pointer
- Pointer Advantages and Disadvantages
- Declaration of Pointer Variables
- Initialization of Pointer Variables
- Dereferencing / Redirecting Pointer Variables
- Declaration versus Redirection
- Void Pointer
- Null Pointer
- Compatibility
- Array of Pointers
- Pointer to Pointer
- Pointer Arithmetic
- Dynamic Memory Allocation Functions

Functions

- Functions
- Advantages of using functions
- Defining a function
- Calling a function
- Return statement
- Function Prototype

- Basic Function Designs
- Programs Using Functions
- Scope
- Recursion
- Iteration vs Recursion
- Nested functions
- Variable Length Number of Arguments
- Parameter Passing Techniques – Call by value & Call by Address
- Functions Returning Pointers
- Pointers and One-Dimensional Arrays
- Pointers and Two-Dimensional Arrays
- Passing 1D arrays to Functions
- Passing 2D arrays to Functions
- Pointers and Strings
- Passing Strings to Functions
- Pointer to Function
- MCQs

Storage Classes

- Object Attributes
- Scope
- Extent
- Linkage
- auto
- static
- extern
- register
- MCQs

Preprocessor Directives

- The #include Preprocessor Directive & User defined header files
- The #define Preprocessor Directive: Symbolic Constants
- The #define Preprocessor Directive: Macros
- Conditional Compilation Directives
- #if

- #else
- #elif
- #endif
- #ifdef
- #ifndef
- #undef
- #error
- #line
- #pragma
- MCQs

Structures, Unions, Enumerations and Typedef

- Structures
- Structure Type Declaration
- Structure Variable Declaration
- Initialization of Structure
- Accessing the members of a structure
- Programs Using Structures
- Operations on Structures (Copying and Comparing Structures)
- Nested structures (Complex Structures)
- Structures Containing Arrays (Complex Structures)
- Array of Structures (Complex Structures)
- Pointer to Structure
- Accessing structure member through pointer using dynamic memory allocation
- Pointers within Structures
- Self-referential structures
- Passing Structures to Functions
- Functions returning Structures
- Unions
- Differences between Structures & Unions
- Enumerated Types / enum keyword
- The Type Definition / typedef keyword
- Bit fields
- MCQs

Command Line Arguments
Files

- Concept of a file
- Streams
- Text File and Binary Files
- State of a File
- Opening and Closing Files
- File Input / Output Functions
- Formatted Input-Output Functions
- Character Input-Output Functions
- Line Input-Output Functions
- Block Input-Output Functions
- File Status Functions (Error Handling)
- Positioning Functions
- System File Operations
- MCQs

Graphics

- Initialization of graphics
- Drawing shapes using pre-defined functions
- Finding the resolution of screen
- Setting colors to text and window
- Font settings
- Fill styles
- Basic GUI applications

C++ Syllabus

Basics of C++

- Introduction to C++
- Different paradigms of problem solving
- POP vs OOP

- Features of Object Oriented Programming Languages
  - Object
  - Class
  - Abstraction
  - Encapsulation
  - Inheritance
  - Polymorphism
  - Dynamic Binding
  - Message Communication
- Constants
- Variables
- Keywords
- Data types
- Declaration of Variables
- Output Stream (cout) & Manipulators
- Input Stream (cin)
- Comments
- Operators
  - Arithmetic operators
  - Relational operators
  - Logical operators
  - Assignment operators & compound assessment operations
  - Increment & decrement operators
  - Conditional operators
  - Bitwise operators
  - Shift operators
  - Type casting
  - Compound assignment operators
  - Address operators
  - Comma operator
  - Pointer operator
  - Sizeof operator
  - new operator
  - delete operator
- Control Statements
- Conditional Control Statements
  - If, if-else

    - nested if-else, if-else-if ladder
- Multiple Branching Control Structure
    - switch-case
- Loop Control statements
    - while
    - do-while
    - for
- Nested Loops
- Jump Control structures
- break
- continue
- goto
- return
- Arrays
- Strings
- Structures
- Pointers
- Dynamic memory allocation using new and delete

Functions

- Defining a Function
- Calling a Function
- Return statement
- Function Prototype
- Basic Function Designs
- Scope
- Reference variables
- Recursion
- Parameter Passing Methods
    - Call by value
    - Call by address
    - Call by reference
- Function Overloading
- Default Arguments
- Inline Functions

Classes and Objects

- Defining a Class
- Creating Objects
- Access specifiers
- Accessing Class Members
- Scope Resolution Operator (::)
- Defining Member Functions
    - Outside the class
    - Inside the class
- Member function with argument
- This pointer
- Passing Objects as Arguments
- Returning Objects
- Array of objects
- Pointer to object
- Dynamic objects
- Friend Functions
- Friend Class
    - Composition
        - Container class
        - Contained class
    - Programs
    - Student Class
    - Employee Class
    - Complex Class
    - Matrix Class
    - Rational Class
    - Circle Class
    - Rectangle Class

Constructors & Destructors

- Constructors
- Properties of constructors
- Types of constructors
    - Default Constructors
    - Parameterized Constructors

    - Copy Constructors
- Constructor Overloading
- Constructors with Default Arguments
- Destructors
- Differences between Member functions & Constructors
- Differences between Constructors & Destructors
- Static Data Members
- Static member functions
- Constant data members
- Constant Member Functions

Operator Overloading

- Defining Operator Overloading Function
- Overloading Unary Operators
- Overloading Binary Operators
- Overloading Unary Operators using Friend Functions
- Overloading Binary Operators using Friend Functions
- Overloading << & >>
- Programs

Inheritance

- Class hierarchies
- Base classes
- Derived Classes
- Derived Class Definition
- Access specifier : protected
- Types of Inheritance & Programs
    - Single inheritance
    - Multiple inheritance
    - Hierarchical inheritance
    - Multi-level inheritance
    - Hybrid inheritance
    - Multi-path inheritance
- Constructors in Derived Classes
- Destructors in Derived Classes

Polymorphism and Virtual Functions

- Static Binding
- Dynamic Binding
- Virtual Destructor
- Function Overriding
- Accessing Members using Pointers
- Virtual Functions
- Pure Virtual Functions
- Abstract Classes
- Virtual Destructors

Templates

- Introduction
- Advantages
- Function Templates
- Over loading function template
- Class Templates
- Inheritance Class Templates

Exception Handling

- Types of Errors
- Benefits of exception handling
- try, catch, throw keywords
- Throwing an exception
- ‘try’ block
- Catching an exception
- Exception objects
- Rethrowing an exception
- Exception Handling Mechanism
- Catching all exceptions
- Nested try blocks

Files

- File Streams Classes
- Opening & Closing a File

- Detection End of File
- File Pointers & Their Manipulation
- Sequential Files
- Random Access Files

I-O Streams

- I-O stream Class hierarchies
- Unformatted I-O Operation
  - get(), put(), getline()
  - write()
  - in cout
  - cin
- Formatted I-O Operations
  - width(), precision()
  - fill(), setf()
  - unsetf()
- Manipulators
  - Manipulator operators
    - Endl, ends
  - manipulator functions
    - setw(), setfill()
    - setprecision()
    - setiosflags()
    - setbase()
    - resetiosflags()
  - User defined manipulators
  - Operator and Overloading

Standard Template Libraries

- Containers
- vector
- list, deque
- arrays
- forward_list
- queue
- priority_queue

- stack
- set, multiset
- map, multimap
- Algorithms
    - Sorting, Searching
    - Important STL Algorithms
    - Useful Array algorithms
    - Partition Operations
- Iterators

JAVA Syllabus

Java Language, OOPS, Programming

- Introduction to Java and OOPS
- Java Tokens- Comments, Identifiers, Keywords, Separators
- Working with Java Editor Software – Editplus, NetBeans, Eclipse
- Packages with static imports
- Working with jar
- Modifiers – File level, Access level and Non-access level
- Datatypes, Literals, Variables, Type Conversion, Casting & Promotion
- Reading runtime values from keyboard and Properties File
- Operators and Control Statements
- Method and Types of methods
- Variable and Types of Variables
- Constructor and Types of constructors
- Block and Types of Blocks
- Declarations, Invocations and Executions
- Compiler & JVM Architecture with Reflection API
- Static Members and their execution control flow
- Non-Static Members and their execution control flow
- Final Variables and their rules

- Classes and Types of classes
- OOPS- Fundamentals, Models, Relations and Principles
- Coupling and Cohesion (MVC and LCRP Architectures)
- Types of objects & Garbage Collection
- Arrays and Var-arg types
- Enum and Annotation
- Design Patterns

Java API and Project

- API and API Documentation
- Fundamental Classes – Object, Class, System, Runtime
- String Handling
- Exception Handling and Assertions
- Multithreading with JVM Architecture
- IO Streams (File IO)
- Networking (Socket Programming)
- Wrapper Classes with Auto boxing and unboxing
- Collections with Generics
- Java 5, 6, 7, 8 new features
- Inner classes
- AWT, Swings, Applet
- Regular Expressions
- Formatting date

PYTHON Syllabus

Core Python

Introduction to Languages

- What is Language?
- Types of languages
- Introduction to Translators
    - Compiler

    - Interpreter
- What is Scripting Language?
- Types of Script
- Programming Languages v/s Scripting Languages
- Difference between Scripting and Programming languages
- What is programming paradigm?
- Procedural programming paradigm
- Object Oriented Programming paradigm

Introduction to Python

- What is Python?
- WHY PYTHON?
- History
- Features – Dynamic, Interpreted, Object oriented, Embeddable, Extensible, Large standard libraries, Free and Open source
- Why Python is General Language?
- Limitations of Python
- What is PSF?
- Python implementations
- Python applications
- Python versions
- PYTHON IN REALTIME INDUSTRY
- Difference between Python 2.x and 3.x
- Difference between Python 3.7 and 3.8
- Software Development Architectures

Python Software’s

- Python Distributions
- Download &Python Installation Process in Windows, Unix, Linux and Mac
- Online Python IDLE
- Python Real-time IDEs like Spyder, Jupyter Note Book, PyCharm, Rodeo, Visual Studio Code, ATOM, PyDevetc

Python Language Fundamentals

- Python Implementation Alternatives/Flavors

- Keywords
- Identifiers
- Constants / Literals
- Data types
- Python VS JAVA
- Python Syntax

Different Modes of Python

- Interactive Mode
- Scripting Mode
- Programming Elements
- Structure of Python program
- First Python Application
- Comments in Python
- Python file extensions
- Setting Path in Windows
- Edit and Run python program without IDE
- Edit and Run python program using IDEs
- INSIDE PYTHON
- Programmers View of Interpreter
- Inside INTERPRETER
- What is Byte Code in PYTHON?
- Python Debugger

Python Variables

- bytes Data Type
- byte array
- String Formatting in Python
- Math, Random, Secrets Modules
- Introduction
- Initialization of variables
- Local variables
- Global variables
- ‘global’ keyword
- Input and Output operations

- Data conversion functions – int(), float(), complex(), str(), chr(), ord()

Operators

- Arithmetic Operators
- Comparison Operators
- Python Assignment Operators
- Logical Operators
- Bitwise Operators
- Shift operators
- Membership Operators
- Identity Operators
- Ternary Operator
- Operator precedence
- Difference between “is” vs “==”

Input & Output Operators

- Print
- Input
- Command-line arguments

Control Statements

- Conditional control statements
- If
- If-else
- If-elif-else
- Nested-if
- Loop control statements
- for
- while
- Nested loops
- Branching statements
- Break
- Continue
- Pass
- Return

- Case studies

Data Structures or Collections

- Introduction
- Importance of Data structures
- Applications of Data structures
- Types of Collections
- Sequence
- Strings, List, Tuple, range
- Non sequence
- Set, Frozen set, Dictionary
- **Strings**
- What is string
- Representation of Strings
- Processing elements using indexing
- Processing elements using Iterators
- Manipulation of String using Indexing and Slicing
- String operators
- Methods of String object
- String Formatting
- String functions
- String Immutability
- Case studies

List Collection

- What is List
- Need of List collection
- Different ways of creating List
- List comprehension
- List indices
- Processing elements of List through Indexing and Slicing
- List object methods
- List is Mutable
- Mutable and Immutable elements of List
- Nested Lists
- List_of_lists

- Hardcopy, shallowCopy and DeepCopy
- zip() in Python
- How to unzip?
- Python Arrays:
- Case studies

Tuple Collection

- What is tuple?
- Different ways of creating Tuple
- Method of Tuple object
- Tuple is Immutable
- Mutable and Immutable elements of Tuple
- Process tuple through Indexing and Slicing
- List v/s Tuple
- Case studies

Set Collection

- What is set?
- Different ways of creating set
- Difference between list and set
- Iteration Over Sets
- Accessing elements of set
- Python Set Methods
- Python Set Operations
- Union of sets
- functions and methods of set
- Python Frozen set
- Difference between set and frozenset ?
- Case study

Dictionary Collection

- What is dictionary?
- Difference between list, set and dictionary
- How to create a dictionary?
- PYTHON HASHING?
- Accessing values of dictionary

- Python Dictionary Methods
- Copying dictionary
- Updating Dictionary
- Reading keys from Dictionary
- Reading values from Dictionary
- Reading items from Dictionary
- Delete Keys from the dictionary
- Sorting the Dictionary
- Python Dictionary Functions and methods
- Dictionary comprehension

Functions

- What is Function?
- Advantages of functions
- Syntax and Writing function
- Calling or Invoking function
- Classification of Functions
    - No arguments and No return values
    - With arguments and No return values
    - With arguments and With return values
    - No arguments and With return values
    - Recursion
- Python argument type functions :
    - Default argument functions
    - Required(Positional) arguments function
    - Keyword arguments function
    - Variable arguments functions
- ‘pass’ keyword in functions
- Lambda functions/Anonymous functions
    - map()
    - filter()
    - reduce()
- Nested functions
- Non local variables, global variables
- Closures
- Decorators
- Generators

- Iterators
- Monkey patching

Advanced Python

Python Modules

- Importance of modular programming
- What is module
- Types of Modules – Pre defined, User defined.
- User defined modules creation
- Functions based modules
- Class based modules
- Connecting modules
- Import module
- From … import
- Module alias / Renaming module
- Built In properties of module

Packages

- Organizing python project into packages
- Types of packages – pre defined, user defined.
- Package v/s Folder
- py file
- Importing package
- PIP
- Introduction to PIP
- Installing PIP
- Installing Python packages
- Un installing Python packages

OOPs

- Procedural v/s Object oriented programming
- Principles of OOP – Encapsulation , Abstraction (Data Hiding)
- Classes and Objects
- How to define class in python
- Types of variables – instance variables, class variables.

- Types of methods – instance methods, class method, static method
-
- Object initialization
- ‘self’ reference variable
- ‘cls’ reference variable
- Access modifiers – private(__) , protected(_), public
- AT property class
- Property() object
- Creating object properties using setaltr, getaltr functions
- Encapsulation(Data Binding)
- What is polymorphism?
- Overriding

1. i) Method overriding
2. ii) Constructor overriding

- Overloading

1. i) Method Overloading
2. ii) Constructor Overloading

iii) Operator Overloading

- Class re-usability
- Composition
- Aggregation
- Inheritance – single , multi level, multiple, hierarchical and hybrid inheritance and Diamond inheritance
- Constructors in inheritance
- Object class
- super()
- Runtime polymorphism
- Method overriding
- Method resolution order(MRO)
- Method overriding in Multiple inheritance and Hybrid Inheritance
- Duck typing
- Concrete Methods in Abstract Base Classes

- Difference between Abstraction & Encapsulation
- Inner classes
- Introduction
- Writing inner class
- Accessing class level members of inner class
- Accessing object level members of inner class
- Local inner classes
- Complex inner classes
- Case studies

Exception Handling & Types of Errors

- What is Exception?
- Why exception handling?
- Syntax error v/s Runtime error
- Exception codes – AttributeError, ValueError, IndexError, TypeError…
    - Handling exception – try except block
    - Try with multi except
    - Handling multiple exceptions with single except block
- Finally block
    - Try-except-finally
    - Try with finally
    - Case study of finally block
- Raise keyword
    - Custom exceptions / User defined exceptions
    - Need to Custom exceptions
- Case studies

Regular expressions

- Understanding regular expressions
- String v/s Regular expression string
- “re” module functions
- Match()
- Search()
- Split()
- Findall()

- Compile()
- Sub()
- Subn()
- Expressions using operators and symbols
- Simple character matches
- Special characters
- Character classes
- Mobile number extraction
- Mail extraction
- Different Mail ID patterns
- Data extraction
- Password extraction
- URL extraction
- Vehicle number extraction
- Case study

File &Directory handling

- Introduction to files
- Opening file
- File modes
- Reading data from file
- Writing data into file
- Appending data into file
- Line count in File
- CSV module
- Creating CSV file
- Reading from CSV file
- Writing into CSV file
- Object serialization – pickle module
- XML parsing
- JSON parsing

Python Logging

- Logging Levels
- implement Logging
- Configure Log File in over writing Mode

- Timestamp in the Log Messages
- Python Program Exceptions to the Log File
- Requirement of Our Own Customized Logger
- Features of Customized Logger

Date & Time module

- How to use Date & Date Time class
- How to use Time Delta object
- Formatting Date and Time
- Calendar module
- Text calendar
- HTML calendar

OS module

- Shell script commands
- Various OS operations in Python
- Python file system shell methods
- Creating files and directories
- Removing files and directories
- Shutdown and Restart system
- Renaming files and directories
- Executing system commands

Multi-threading & Multi Processing

- Introduction
- Multi tasking v/s Multi threading
- Threading module
- Creating thread – inheriting Thread class , Using callable object
- Life cycle of thread
- Single threaded application
- Multi threaded application
- Can we call run() directly?
- Need to start() method
- Sleep()
- Join()
- Synchronization – Lock class – acquire(), release() functions

- Case studies

Garbage collection

- Introduction
- Importance of Manual garbage collection
- Self reference objects garbage collection
- ‘gc’ module
- Collect() method
- Threshold function
- Case studies

Python Data Base Communications(PDBC)

- Introduction to DBMS applications
- File system v/s DBMS
- Communicating with MySQL
- Python – MySQL connector
- connector module
- connect() method
- Oracle Database
- Install cx_Oracle
- Cursor Object methods
- execute() method
- executeMany() method
- fetchone()
- fetchmany()
- fetchall()
- Static queries v/s Dynamic queries
- Transaction management
- Case studies

Python – Network Programming

- What is Sockets?
- What is Socket Programming?
- The socket Module
- Server Socket Methods
- Connecting to a server

- A simple server-client program
- Server
- Client

Tkinter & Turtle

- Introduction to GUI programming
- Tkinter module
- Tk class
- Components / Widgets
- Label , Entry , Button , Combo, Radio
- Types of Layouts
- Handling events
- Widgets properties
- Case studies

Data analytics modules

- Numpy
- Introduction
- Scipy
- Introduction
- Arrays
- Datatypes
- Matrices
- N dimension arrays
- Indexing and Slicing
- Pandas
- Introduction
- Data Frames
- Merge , Join, Concat
- MatPlotLib introduction
- Drawing plots
- Introduction to Machine learning
- Types of Machine Learning?
- Introduction to Data science

DJANGO

- Introduction to PYTHON Django
- What is Web framework?
- Why Frameworks?
- Define MVT Design Pattern
- Difference between MVC and MVT