



Softcrayons

C & C++ Programming Language

*Empowering minds
shaping futures*



PROFESSIONAL CERTIFICATION IN

C & C++ PROGRAMMING LANGUAGE



PROSPECTUS



Key Advantages of Choosing Softcrayons

FOR STUDENTS

Supplemental Learning Resources: Softcrayons offers offline and online courses, educational materials, and additional resources that can complement and enhance college students' learning.

Skill Development: Softcrayons offer courses and certifications focused on developing specific skills that are in high demand in the job market.

Career Exploration: Softcrayons offer a wide range of courses across various disciplines, enabling college students to explore different fields and potential career paths.

Industry Relevance: Softcrayons frequently collaborates with industry professionals and experts to ensure that the knowledge and skills imparted are relevant and aligned with current industry practices and trends.

FOR FREELANCER & JOB SEEKER

Flexibility: Softcrayons offers online courses and programs that can be accessed from anywhere.

Skill Development: To acquire in-demand skills according to the latest industry trends and technologies to stay competitive in the job market.

Key Advantages of Choosing Softcrayons

Certifications: Softcrayons provides you with Professional Certifications and helps you with Resume Enhancement.

Career Support: Softcrayons also offers career counseling and job placement assistance, which can be invaluable for freelancers seeking new projects or job seekers looking for employment.

FOR ENTREPRENEURS AND BUSINESS OWNER

Upskilling and Reskilling: As the business landscape evolves rapidly, Softcrayons ensure that you stay up-to-date with the latest trends, technologies, and best practices.

Flexible Learning: Online courses offered by Softcrayons allow you to learn at your own pace, fitting your studies around your busy schedules.

Entrepreneurial Skills Development: Softcrayons offers courses and programs specifically designed to help entrepreneurs develop essential skills.

Cost-effective: Softcrayons provides more affordable learning options that help you invest in your professional development without straining your budget.

About The Program



C and C++ are foundational programming languages crucial to software development. C, developed by Dennis Ritchie in the early 1970s, is a procedural language known for its efficiency and control over system resources, making it ideal for operating systems, embedded systems, and performance-critical applications. It emphasizes direct manipulation of memory through pointers and provides low-level access to hardware.

C++, created by Bjarne Stroustrup in the 1980s, extends C with object-oriented features, enabling more complex and scalable software design. It introduces concepts like classes, inheritance, and polymorphism, which facilitate code reuse and abstraction. C++ maintains C's performance advantages while adding support for generic programming through templates and the Standard Template Library (STL). Both languages are widely used: C for system-level programming and C++ for applications requiring high performance and modularity. Mastery of C and C++ provides a deep understanding of computer science principles and efficient software development practices.



C & C++

TRAINING CURRICULUM

1 Fundamentals in C

- What is 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

2 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

3 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

4 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

5 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

6 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

7 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

8 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

9 Storage Classes

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

10 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**

11 **Command Line**

12 **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**

13 **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**

14 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 assignment 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

15 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

16 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
 - Programs
 - Student Class
 - Employee Class
 - Complex Class
 - Matrix Class
 - Rational Class
 - Circle Class
 - Rectangle Class

17 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

18 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

19 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

20 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

21 Templates

- Introduction
- Advantages
- Function Templates
- Overloading function template
- Class Templates
- Inheritance Class Templates

22 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

23 Files

- File Streams Classes
- Opening & Closing a File
- Detection End of File
- File Pointers & Their Manipulation
- Sequential Files
- Random Access Files

24 I-O Streams

25 I-O stream Class hierarchies

26 Unformatted I-O Operation

27 Formatted I-O Operations

— MANIPULATORS

- Manipulator operators
- User defined manipulators
- Operator and Overloading

- **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**

28 **Standard Template Libraries**

- CONTAINERS
 - VECTOR
 - LIST, DEQUE
 - ARRAYS
 - FORWARD_LIST
 - QUEUE
 - PRIORITY_QUEUE
 - STACK
 - SET, MULTISSET
 - MAP, MULTIMAP
 - ALGORITHMS
- **Sorting, Searching**
 - **Important STL Algorithms**
 - **Useful Array algorithms**
 - **Partition Operations**

PLACEMENT COMPANIES



Testimonials of Students



Sukhpreet Kaur

2 reviews

★★★★★ 2 weeks ago **NEW**

Hlo mam I'm Sukhpreet your Softcrayons tarining institute in student my training is digital marketing course 😊 I'm very becoz I'm digital marketing beginner but my experience to much becoz my trainer is very intelligent and supportive and nature is very friendly 😊



Manish Malik

1 review

★★★★★ 2 weeks ago **NEW**

I'm new student in softcrayons my starting classes all gud my softcrayons experience to much better becoz my trainer is very experienced



Aman Bhardwaj

2 reviews

★★★★★ 5 months ago

I am Aman Bhardwaj, Recently I completed a Digital Marketing course from Softcrayons. After completing my course I got a placement at SNVA Ventures with a good salary package. If you want to do a course and boost your career in the Digital Marketing field. I will recommend you visit Softcrayons. If I talk about the environment and faculty then Softcrayons have a very good and friendly environment and their faculty is highly experienced in the Digital Marketing field. Specifically, Yashvant sir is one of the best trainer and they have great experience in the Digital Marketing field. Thank you Softcrayons and all staff who helped me boost my career in the Digital Marketing field.



Shivam Sharma

1 review

★★★★★ 2 months ago

I got the chance to study with the best teacher and they provided me a good career guidance. a veryb great place to learn programming and start your career.



Lalita Tiwari

1 review

★★★★★ 5 months ago

I hearded about softcrayons through friends and I enroll myself here, and done my course. I suggest you all to join softcrayons. Hope you do great.



Tanish Chandrawal

5 reviews

★★★★★ 5 months ago

It is good institute, practical oriented practice is very good. This institute is very useful for graduate students to make carrier in IT. 100% job guarantee is available for all students. Very good Institute for Cloud Computing like Azure, AWS, GCP.



Aman Vishwakarma

2 reviews

★★★★★ 2 months ago

Hie guys I'm aman softcrayons institute students for AutoCAD.. softcrayons institute is best training institute sarfaraz sir is best teacher for softcrayons. And best institute softcrayons



SOFTCRAYONS®



GAZIABAD

693, Sector 14-A, Vasundhara,
Ghaziabad, UP (201010)

NOIDA

B-132, Sector 2, Near Sector 15
Metro Station, Noida UP (201301)