

### Skill Enhancement Courses (SEC)

Course Code	Course Name	Level	L	T	P	C	CIE	SEE	Total	Pre-requisite
241AI033	Programming with C++	IC			2	2	50	50	100	-
241CS017	Object Oriented Analysis & Design using UML	IC			2	2	50	50	100	ASE
241AI034	Web Application Development using MERN Stack	AC			2	2	50	50	100	IMSD
241IT008	Bigdata Spark	AC			2	2	50	50	100	-
241CS019	CI/CD using DevOps	AC			1	1	100	-	100	-
<b>Total</b>					<b>9</b>	<b>9</b>				

## PROGRAMMING with C++

Course Code: 241AI033

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>2</b>	<b>2</b>

### Course Outcomes:

At the end of the course, student will be able to:

- CO1:** Apply C++ programming concepts recursion, functions, and scope resolution in problem-solving.
- CO2:** Design programs using classes, encapsulation, access specifiers, constructors, destructors, and operator overloading
- CO3:** Analyse forms of inheritance and polymorphism to build flexible and reusable code components.
- CO4:** Utilize C++ features templates, exception handling, and standard template library (STL) containers to create generic, robust, and high-performance applications.
- CO5:** Develop a deep understanding of memory management, pointers, and virtual base classes to solve problems involving complex inheritance and runtime behaviours in C++.

### Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2	2	1	1				1		1
CO2	3	3	2	1	2				1		1
CO3	3	2	3	2	1						1
CO4	3	3	2	2	3				1		1
CO5	3	3	3	2	3						1

### Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	
CO2	2	
CO3	2	
CO4	2	
CO5	2	

### Practice:

1. Fundamental concepts of C++
  - a. **Concepts Covered:** Recursion is a process  
Develop a C++ Program to Find Factorial of a Given Number Using Recursion.
  - b. **Concepts Covered:** arguments to a function, Call by Value and Call by Reference.  
Develop a C++ Program to Demonstrate Call by Value and Call by Reference

2.

- a. Scope Resolution and Namespaces

**Concepts Covered:** Global vs Local variable, Scope resolution operator, Declaring and using custom namespaces, using directive vs fully qualified names

Develop a C++ program that demonstrates the use of the scope resolution operator and namespaces.

- b. Inline Functions

**Concepts Covered:** Definition, Benefits and Syntax of inline functions, Situations where inline functions are not recommended

Create a C++ program that illustrates the use of inline functions

3.

- a. classes, objects, and encapsulation

**Concepts Covered:** Defining and creating a class, Declaring objects, Data members and member functions, Basic encapsulation principles

Develop a C++ program that models a Bank Account using a class. The class include data members account number, name, balance and member functions deposit, withdraw, display balance.

- b. Access Specifiers: public and private

**Concepts Covered:** Access specifiers, Data hiding and encapsulation, Member access control

Create a C++ program that illustrates the difference between the public and private access specifiers.

- c. this Pointer

**Concepts Covered:** this pointer, Resolving naming conflicts, Returning the current object.

Develop a C++ program that uses the this pointer to refer to the current object

4.

- a. Function Overloading

**Concepts Covered:** Function overloading, Function signatures, Use cases of function overloading

Create a C++ program that demonstrates function overloading by defining multiple functions with the same name but different parameter types or counts.

- b. Default Arguments in Functions

**Concepts Covered:** Syntax for default parameters, Rules and limitations of default arguments, Function call variations using default values

Develop a C++ program that illustrates the use of default arguments in functions.

- c. Friend Function

**Concepts Covered:** Friend functions and their declaration, Accessing private members from non-member functions, Controlled access and encapsulation

Create a C++ program that uses a friend function to access the private data of a class

5.

- a. Constructors and Destructors

**Concepts Covered:** Default constructor, Destructor, Constructor and destructor invocation timing, Object lifecycle management

Create a C++ program that demonstrates the use of constructors and destructors in a class.

- b. Constructor Overloading

**Concepts Covered:** Constructor overloading, Different ways of object initialization, Function overloading principles applied to constructors  
Develop a C++ program that illustrates constructor overloading.

c. Copy Constructor

**Concepts Covered:** Copy constructor syntax and use, Passing objects by value, When the copy constructor is invoked  
Write a C++ program that illustrates the use of a copy constructor

6.

a. Overloading Unary and Binary Operators using Member Functions

**Concepts Covered:** Operator overloading syntax, Overloading unary operators and binary operators, Importance of return types and chaining  
Develop a C++ program that demonstrates how to overload both unary and binary operators using member functions.

b. Overloading Unary and Binary Operators using Friend Functions

**Concepts Covered:** Syntax and declaration of friend functions, Accessing private members outside the class, Difference between member and friend function, Operator overloading with friend functions.

Create a C++ program to demonstrate operator overloading for unary and binary operators using friend functions

7.

a. Exploring Various Forms of Inheritance in C++

**Concepts Covered:** Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance, Hybrid Inheritance  
Develop C++ programs to demonstrate different forms of inheritance

b. Order of Execution of Constructors and Destructors in Inheritance

**Concepts Covered:** Constructor execution order, Destructor execution order, Constructor and destructor chaining

Develop a C++ program that illustrates the order of execution for constructors and destructors in the context of inheritance.

8.

a. Illustrating Pointers to a Class

**Concepts Covered:** Pointers to objects, Dereferencing and member access, Memory management

Develop a C++ program that demonstrates how to use pointers to access and manipulate objects of a class.

b. Illustrating Virtual Base Class

**Concepts Covered:** Diamond problem, Virtual base class, Order of constructor and destructor calls

Develop a C++ program to demonstrate the concept of virtual base classes in the context of multiple inheritance, which resolves ambiguity in the inheritance hierarchy

9.

a. Virtual Functions in C++

**Concepts Covered:** Virtual functions, Function overriding, Dynamic binding  
Develop a C++ program to demonstrate the use of virtual functions to achieve dynamic dispatch and enable runtime polymorphism.

b. Runtime Polymorphism in C++

**Concepts Covered:** Runtime polymorphism, Base class pointer/reference, Overriding methods

Develop a C++ program that illustrates runtime polymorphism using virtual functions

10.

- a. Function Templates in C++

**Concepts Covered:** Function template, Template argument deduction

Develop a C++ program that demonstrates the use of function templates to create functions that can work with different data types.

- b. Template Classes in C++

**Concepts Covered:** Template class, Code reusability

Develop a C++ program that demonstrates template classes, which allow creating classes that can work with any data type

11.

- a. Handling Exceptions in C++

**Concepts Covered:** Exception handling mechanism, try block, throw statement, catch block.

Develop a C++ program that demonstrates exception handling using try, throw, and catch blocks.

- b. Using Multiple Catch Statements in C++

**Concepts Covered:** Multiple catch blocks, Handling various exception types, Exception hierarchy

Develop a C++ program to illustrate the use of multiple catch statements, where different types of exceptions are caught and handled differently.

12.

- a. Implementing List, Vector, and their Operations

**Key Concepts:** STL, List and Vector Operations

Develop a C++ program to implement List and Vector containers and perform basic operations such as insertion, deletion, traversal.

- b. Implementing Deque and its Operations

**Key Concepts:** Deque operations

Implement Deque in C++ and demonstrate basic operations.

- c. Implementing Map and Map Operations

**Key Concepts:** Map operations

Implement Map and demonstrate operations such as insertion, deletion, access, and searching

#### **Additional Practice:**

1. Develop a C++ Program for Flight Booking System
2. Develop a Qt Application Containing Slider and Spin Box(Slider Responds to Spin Box Changes)
3. Develop a Qt Application for Creating a Text Pad
4. C++ Program for a Guessing Game with Asterisks

This program presents a guessing game where the user must guess letters of a mystery word, represented initially by asterisks. The program allows 3 incorrect guesses. After each incorrect guess, the remaining chances are displayed.

A program with maximum of 20 characters, user will be guessed and will show only asterisks (\*) on the screen.

The user will input one character at a time. And for every correct character, the asterisk will be replaced by that character until all the characters or the mystery word/s will reveal.

Your program will accept a maximum three (3) errors or mistakes in entering/inputting character otherwise the mystery word/s will be viewed.

Sample Output: Output: \*\*\*\*\*

Enter your character: e    Output: \*\*\*e\*\*e

Enter your character: a    Output: sorry! the character is not existing. you still have 2 chances

Enter your character: s    Output: s\*\*e\*\*e

Enter your character: c    Output: sc\*e\*ce

Enter your character: i    Output: scie\*ce

Enter your character: n    Output: science

### Reference Books:

- 1 C++ Primer Plus by Stephen Prata, Sixth Edition, Pearson, ISBN: 978-9332546189
- 2 C++ GUI Programming with Qt4, Jasmin Blanchette, Mark Summerfield, Second Edition, Prentice Hall Press, ISBN: 978-0132354165
- 3 C++ for Programmers, Paul J. Deitel, Harvey M. Deitel, Pearson, ISBN: 978-0137001309

### Web Links:

- 1 <http://en.cppreference.com/w/cpp/links/libs>
- 2 <https://www.daniweb.com/digital-media/ui-ux-design/threads/113591/trying-to-run-a-program-through-a-web-link>
- 3 <http://www.yolinux.com/TUTORIALS/LinuxTutorialC++.html>
- 4 <https://github.com/fffaraz/awesome-cpp>
- 5 [http://www.techsystemembedded.com/cpp\\_links.ph](http://www.techsystemembedded.com/cpp_links.ph)