

**Government of Karnataka  
Department of Technical Education  
Board of Technical Examinations, Bengaluru**

Course Title: <b>Object Oriented Programming Using C++</b>	Course Code : <b>15EC63C</b>
Credits : <b>4</b>	Semester : <b>6</b>
Teaching Scheme in Hrs (L:T:P) : <b>4:0:0</b>	Course Group : <b>Elective</b>
Type of course : <b>Lecture</b>	Total Contact Hours : <b>52</b>
CIE : <b>25 Marks</b>	SEE : <b>100 Marks</b>

### Prerequisites

Knowledge of programming and C language.

### Course Objectives

1. To understand the need for high-level languages including C++ and programming paradigms.
2. To understand the syntax of C++ and writing simple programs in C++
3. To understand the need and role of object oriented programming for real-world applications.
4. To enable the students to write simple programs using OOP concepts

### Course Outcomes

On successful completion of the course, the students will be able to

1. Understand the C++ language features.
2. Use the control structure and data types in C++.
3. Write simple programs using classes and objects.
4. Write simple programs to implement overloading concepts.
5. Understand the concepts of inheritance and polymorphism.
6. Understand virtual functions and I/O statements.

Course Outcome		CL	Linked POs	Teaching Hours
CO1	Understand the C++ language features.	R/U/C	1,2,4,10	09
CO2	Use the control structure and data types in C++.	R/U/A	1,2,4,10	10
CO3	Write simple programs using classes and objects.	R/U/A	1,2,4,10	09
CO4	Write simple programs to implement overloading concepts.	U/A/C	1,2,4,10	08
CO5	Understand the concepts of inheritance and polymorphism.	U/A/C	1,2,4,10	07
CO6	Understand pointers, virtual functions and I/O statements.	U/A	1,2,10	09
<b>Total Sessions</b>				<b>52</b>

**Legends:** PO-Program Outcome, CO-Course Outcome, CL-Cognitive Level, R-Remember, U-Understand, A-Apply, C-Create

## Mapping Course Outcomes with Program Outcomes

Course Outcomes	Programme Outcomes									
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	*	*	--	*	--	--	--	--	--	*
CO2	*	*	--	*	--	--	--	--	--	*
CO3	*	*	--	*	--	--	--	--	--	*
CO4	*	*	--	*	--	--	--	--	--	*
CO5	*	*	--	*	--	--	--	--	--	*
CO6	*	*	--	---	--	--	--	--	--	*

Legend: \* Linked, -- No link

## Course-PO attainment matrix

Course	Programme Outcomes									
	1	2	3	4	5	6	7	8	9	10
Object Oriented Programming using C++	3	3	--	3	--	--	--	--	--	3

Legend: Addressing levels: 1-Slight, 2-Moderate, 3-Substantial, -- Not addressed

**Quantification Method:** This is to relate the level of PO with the number of hours devoted to the COs which address the given PO. If  $\geq 40\%$  of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 3; if 25 to 40%, Level 2; if 5 to 25%, Level 1; and if  $< 5\%$ , not addressed.

## Course content and pattern of marks for SEE

Unit	Unit Name	Teaching Hours	Weightage for CLs in SEE			Marks	Weightage (%)
			R	U	A		
1	Introduction to OOP and C++	9	05	10	10	25	12
2	Control structure and data types.	10	05	05	10	20	14
3	Classes and objects.	09	05	10	10	25	25
4	Overloading.	08	05	10	10	25	15
5	Inheritance, polymorphism.	07	05	05	15	25	15
6	Pointers, Virtual functions, console I/O statements.	09	05	05	15	25	19
<b>Total</b>		<b>52</b>	<b>30</b>	<b>45</b>	<b>70</b>	<b>145</b>	<b>100</b>

Legend: R- Remember, U-Understand A-Application

## Course Contents

### Unit – 1: Introduction to OOP and C++

**Duration: 09 Hrs.**

Features of C language, POP and OOP. List of OOP languages. Basic format of C++ program. Processor Directives. I/O statements. Language syntax: Keywords, identifiers, constants, variables, classification of variables based on scope and life, operators. Data types: Basic, derived and user-defined data types. Data-type casting. Data abstraction and encapsulation. Simple example programs.

### Unit – 2: Control Structure and Data Types

**Duration: 10 Hrs.**

Concepts of control structure. Branching- if, if-else, switch, break, continue. Looping-for, while and do-while. Derived data type: Arrays, strings, pointers, enumerated data types and functions. Functions-Call by value, address and reference. User-defined data type: structure, union and classes. Example programs.

### Unit – 3: Classes and Objects

**Duration: 09 Hrs.**

Introduction to class and objects. Limitation of Structure and benefits of class. Class and object creation. Private and public, protected members of class-Variables, arrays and functions. Accessing Class Members. Memory allocation for Objects. Array of objects. Friend Function. Data abstraction and data encapsulation. Simple example programs.

### Unit-4: Overloading

**Duration: 08 Hrs.**

Introduction to overloading. Constructors: parameterized constructors, default arguments, overloading and copy constructor. Destructors, Unary and binary operator overloading. Function overloading, functions with default arguments. Inline functions. Simple example programs.

### Unit-5: Inheritance and Polymorphism

**Duration: 07 Hrs.**

Introduction to inheritance. Defining derived classes, Levels of inheritance, Single inheritance, public and private member inheritance, multiple inheritance, Hierarchical inheritance, Hybrid inheritance, polymorphism. Example programs.

### Unit-6: Pointers, Virtual Functions, Console I/O Statements

**Duration: 09 Hrs.**

Pointers-Declaration and initialization. Manipulation of pointers, pointers to objects, pointers to derived classes, pointers with arrays and strings. Introduction to Virtual functions, rules for virtual functions, pure virtual functions. Formatted and Unformatted I/O functions. Simple example programs. Features of other OOPS languages-JAVA and PYTHON.

## References

1. Object oriented programming with C++--Rabertlafer
2. Object oriented programming with C++-- 4E, E Balaguruswamy, Tata McGraw hill
3. OOPS with C++ By Niranjan A., Sapna Publications
4. Object oriented programming in C++, By P B Kottur., Sapna Publications

5. Object oriented programming in C++, Dr.G.T. Thampi, Dr. S.s.Matha,Dreamtech, 2009 edition,
6. Object oriented programming in C++, Rajesh K Shukla ,Wiely Precise text Book.2008.
7. Object Oriented Programming with C++ , SouravSahay, Oxford Higher Education
8. Analysis & Designing Of Algorithms with C/C++ -By Nandagopalan.
9. C++ Complete Reference, Herbert Schilt. TMH.
10. Programming in C++ , M T Somashekar , PHI
11. Professional C++ , Wiley India (Wrox )

## Course Delivery

The course will be delivered through lectures, presentations and support of modern tools. Student activities are off-class.

## Course Assessment and Evaluation Scheme

### Master Scheme

Assessment Method	What		To Whom	Assessment mode /Frequency /timing	Max. Marks	Evidence Collected	Course Outcomes
Direct assessment	CIE	IA	Students	Three tests <sup>+</sup>	20	Blue Books	1 to 6
				Activity*	05	Activity Sheets	1 to 6
	SEE	End exam		End of the course	100	Answer Scripts at BTE	1 to 6
				<b>Total</b>	<b>125</b>		
Indirect assessment	Student feedback on course		Students	Middle of the Course	Nil	Feedback Forms	1 to 3 Delivery of course
	End of course survey			End of the Course	Nil	Questionnaires	1 to 6 Effectiveness of delivery instructions & assessment methods

**Legends:** CIE-Continuous Internal Evaluation, SEE- Semester End-exam Evaluation

<sup>+</sup> Every I.A. test shall be conducted for 20 marks. Average of three tests, by rounding off any fractional part thereof to next higher integer, shall be considered for IA.

<sup>\*</sup>Students should do activity as per the list of suggested activities/ similar activities with prior approval of the teacher. Activity process must be initiated well in advance so that it can be completed well before the end of the term and assessed through appropriate Rubrics.

Questions for CIE and SEE will be designed to evaluate the various CLs as per the Weightage shown in the following table.

Sl. No.	Cognitive Levels (CL)	Weightage (%)
1	Remembering	15
2	Understanding	30
3	Applying	40

4	Evaluate	10
5	Create	05
<b>Total</b>		<b>100</b>

### Continuous Internal Evaluation (CIE) pattern

#### (i) Student Activity (5 marks)

The following student activities or similar activities can be assigned for assessing CIE/IA marks

Sl. No.	Activity
1	Execute and submit at least two simple programs from each of unit1, unit2 and unit3
2.	Execute and submit at least two programs from each of unit4, unit5 and unit6.
<b>Execution Mode</b>	
<ol style="list-style-type: none"> <li>1. Activity 1 and 2 are mandatory for every batch; every batch can have maximum of 4 students.</li> <li>2. Activities shall be carried out batch-wise throughout the semester and submit one report per batch before the end of the semester.</li> <li>3. Report shall be qualitative and not to exceed 6 pages.</li> <li>4. Each of the activity can be carried out off-class; however, demonstration/presentation should be done in the Lab/class room.</li> <li>5. Teacher is expected to observe and record the progress of students' activities</li> <li>6. Assessment shall be in accordance with the following <b>rubrics</b> table.</li> </ol>	

### Institutional activities (No marks)

The following are suggested institutional activities, to be carried out at least one during the semester. The course teacher/coordinator is expected to maintain the relevant record (Containing, Activity name, Resource persons and their details, duration, venue, student feedback, etc) pertaining to Institutional activities.

Sl. No.	Activity
1	Practice some programs related to topics of the course in the Lab.
2	Arrangement of a talk/seminar/lecture on OOP languages

#### (ii) Model of rubrics for assessing student activity (for every student)

Dimension	Scale					Marks (Example)
	1 Unsatisfactory	2 Developing	3 Satisfactory	4 Good	5 Exemplary	
1. Research and gathering information	Does not collect information relate to topic	Collects very limited information, some relate to topic	Collects basic information, most refer to the topic	Collects more information, most refer to the topic	Collects a great deals of information, all refer to the topic	3
2. Full-fills team roles and duties	Does not perform any duties assigned to the team role	Performs very little duties	Performs nearly all duties	Performs almost all duties	Performs all duties of assigned team roles	2
3. Shares work	Always relies on others to do	Rarely does the assigned	Usually does the	Always does the assigned	Always does the assigned	5

equality	the work	work, often needs reminding	assigned work, rarely needs reminding	work, rarely needs reminding.	work, without needing reminding	
4. Listen to other team mates	Is always talking, never allows anyone to else to speak	Usually does most of the talking, rarely allows others to speak	Listens, but sometimes talk too much,	Listens and talks a little more than needed.	Listens and talks a fare amount	3
<b>Total marks</b>						ceil(13/4)= 4

**(iii) CIE/IA Tests (20 Marks)**

Three tests have to be conducted in accordance with the test pattern given below and average marks of them are considered for CIE/IA with specified schedule. Fractional average marks can be rounded-off to next higher integer.

**(iv) Format of CIE/IA test question paper**

CIE Question Paper					
Institution Name and Code					
Course Coordinator/Teacher					
<i>Program Name</i>		<i>Test No.</i>		<i>Units</i>	
<i>Class/Sem</i>		<i>Date</i>		<i>CL</i>	
<i>Course Name</i>		<i>Time</i>		<i>COs</i>	
<i>Course Code</i>		<i>Max. Marks</i>		<i>POs</i>	
Note to students: Answer all questions					
Question No.	Question	Marks	CL	CO	PO
1					
2					
3					
4					

**Legends:** PO-Program Outcome, CO-Course outcome, CL-Cognitive Level, R-Remember, U-Understand, A-Apply

**Note:** Internal choice may be given in each CO at the same cognitive level (CL).

**(v) Model question paper for CIE**

CIE Question Paper					
Institution Name and Code					
Course Coordinator/Teacher					
<i>Program Name</i>	Electronics and Communication	<i>Test No.</i>	1	<i>Units</i>	1 & 2
<i>Class/Sem</i>	5 <sup>TH</sup> Sem	<i>Date</i>	1/1/2017	<i>CL</i>	R/U/A
<i>Course Name</i>	Object Oriented Programming using C++.	<i>Time</i>	10-11AM	<i>COs</i>	1 & 2
<i>Course Code</i>	15EC54B	<i>Max. Marks</i>	20	<i>POs</i>	1, 2 & 3
Note to students: Answer all questions					
No.	Question	Marks	CL	CO	PO
1	List features of object Oriented Programming OR Identify Keywords, Tokens.	05	U/A	1	1,2,3
2	Classify variables based on Scope and life time	05	A	1	1,2
3	Describe if-else with syntax and example. OR Express Conditional statements With Syntax.	05	U	2	1,2

4	Describe Declaration and Initialization of one dimensional array with syntax and Example	05	A	2	1,2
---	--	----	---	---	-----

### Semester End-exam Evaluation (SEE)

#### (i) End-exam question-paper pattern

Unit	Unit Name	Study Duration (Hrs.)	No. Questions for end-exam	
			PART – A 5 Marks	PART – B 10 Marks
1	Introduction to OOP and C++	9	02	01
2	Control structure and data types .	10	01	02
3	Classes and objects.	09	02	02
4	Overloading.	08	01	02
5	Inheritance, polymorphism	07	02	01
6	Pointers, Virtual functions, console I/O statements.	08	01	02
<b>Total</b>		<b>52</b>	<b>09 (45 Marks)</b>	<b>10 (100 Marks)</b>

#### (ii) Model question paper

Course Title : **Object Oriented Programming using C++**

Course Code : **15EC54B**

Time : **3 Hrs**

Semester : **Fifth**

Max. Marks: **100**

Instructions: 1. Answer any **SIX** question from **Part A** (5x6=30 Marks)

2. Answer any **SEVEN** full questions from **Part B** (7x10=70 Marks)

#### Part A

1. List the features of Object Oriented Programming.
2. Explain processor Directives in C++.
3. Differentiate while and do-while.
4. Demonstrate creating a Classes and objects
5. Express Friend functions Friend Class.
6. Summarize Rules for operator overloading
7. Illustrate polymorphism means in C++.
8. Discuss Hierarchical inheritance.
9. Discuss the need of Virtual function.

#### Part B

1. Explain built-in data types and Derived data types.
2. Describe Declaration and Initialization of two dimensional arrays with syntax and Example.
3. Explain String handling functions.
4. Compute accessing Class Members.
5. Differentiate between private, public and protected.
6. Illustrate Function overloading.
7. Compute some Example programs related to overloading.
8. Explain public and private member inheritance.

- Describe Class Templates, Class Templates with multiple parameters
- Demonstrate to make Virtual function “Pure”. Explain implications making a function a pure virtual function.

## Model Question Bank

**Note:** The questions in the question bank are indicative but not exhaustive. Sub-questions on different CLs may be combined in 10-marks questions or 10-marks questions can be spitted into if necessary keeping Weightage of CLs approximately intact.

### Unit-1 Introduction to OOP and C++ 5-Mark questions

#### Remember

- Describe object oriented programming.
- List advantages of object oriented programming.
- List features of Procedure Oriented Programming.
- List features of object Oriented Programming.
- Identify Keywords, Tokens.
- Tabulate different types of Constants.

#### Understand

- Discuss Identifiers and their naming rules.
- Illustrate variables Declaration and Initialization procedure.
- Discuss Data types.
- Explain processor Directives.
- Differentiate between object oriented programming and procedure oriented programming.

#### Application

- Write a program to read two numbers from the keyboard and display large no. on the screen.
- Write a program to read the values of a, b and c and display the value of x, where  $x=a/b-c$ . Assume some values for a, b, c.
- Write a program calculates the area of a rectangle and displays it.

### 10-Mark questions

#### Understand

- Explain basic structure of C++ program with Examples.
- Summarize I/O statements with syntax and Examples.
- Classify variables based on Scope and life time.
- Discuss Arrays and Strings with syntax and Examples.
- Explain built-in data types and Derived data types.
- Explain Operators in C++.

#### Create

- Write a program to display the following output using a single **cout** statement.  
Maths = 90  
Physics = 77  
Chemistry = 69

### Unit-2 Control structure and data types 5-Mark questions

#### Remember

- List different types of control structures.
- Describe if, if-else with syntax and example
- Define break, continue with syntax and example.



### Understand

1. Represent switch statement with syntax and example.
2. Express Conditional statements With Syntax.
3. Classify array with syntax.
4. Differentiate while and do-while.
5. Differentiate C and C++.

### Application

6. Write a program to display a string entered by user.
7. Write a Program to illustrate Returning structure from function in C++.

### 10-Mark questions

#### Understand

1. Explain for, while, do-while loop control structure with syntax.
2. Describe Declaration and Initialization of one dimensional array with syntax and Example.
3. Describe Declaration and Initialization of two dimensional array with syntax and Example.
4. Compute program for 'for' loop with respect to C++.
5. Compute program for addition of diagonal elements of matrix with C++ statements.
6. Explain String handling functions.

#### Create

7. Write a program to display marks of 5 students by passing one- dimensional array to a function.
8. Write a program to display the following output using **for** loops  
1  
22  
333  
4444  
.....
9. Write a program to evaluate the following function to 0.0001% accuracy  
 $\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$
10. Write a program to Calculate the variance and standard deviation of N numbers.
11. Write a program to illustrate elements of two dimensional array by passing it to a function.
12. Write a C++ program to define enumeration type and assign value to variable of that type.

### Unit-3Classes and Objects

#### 5-Mark questions

#### Remember

1. Define Class and objects.
2. Describe specifying Class.
3. Describe procedure for creating a Classes and objects

#### Understand

1. Discuss limitation of Structure.
2. Discuss instantiation of objects?
3. Distinguish between Object and Class.
4. Experiment with some examples of Classes
5. Report with the help of suitable example how data abstraction and data encapsulation takes place.
6. Explain Friend functions and Friend Class.

### 10-Mark questions

#### Understand

1. Relate Declaration of arrays and function with respect to class and objects.
2. Compute accessing Class Members.

3. Differentiate between private, public and protected.
4. Discuss Memory allocation for Objects.
5. Indicate array of objects.

### **Create**

6. Write a Program to illustrate working of Objects and Class in C++ Programming.
7. Define a class to represent a bank account. Include the following members  
Data members  
1. Name of the depositor. 2. Account Number  
3. Type of account. 4. Balance amount in account.  
Member functions  
1. To assign initial values  
2. To deposit an amount  
3. To withdraw an amount after checking the balance  
4. To display name and balance.  
Write a program to test the program.

## **Unit-4 Overloading 5-Mark questions**

### **Understand**

1. Establish Constructors and Destructors.
2. Represent Parameterized constructors.
3. Describe Constructor overloading.
4. Report Constructors with default arguments.
5. Explain Copy constructor with syntax.
6. Summarize Rules for operator overloading.

### **10-Mark questions**

### **Understand**

1. Describe Operator overloading -- unary and Binary operators
2. Illustrate Function overloading.
3. Describe functions with default arguments.
4. Discuss Inline functions.

### **Create**

5. Compute some Example programs related to overloading.
6. Write a program to return absolute value of variable types integer and float using function overloading.
7. Write a program to add two complex numbers by passing objects to function.
8. Write a program demonstrate the working of overloaded constructors.

## **Unit-5 Inheritance, Polymorphism 5-Mark questions**

### **Remember**

1. Define inheritance.
2. Describe derived classes.

### **Understand**

3. Discuss syntax for Single inheritance.
4. Discover to use protected visibility specifies to class members.
5. Discuss Hierarchical inheritance.
6. Describe how an object of a class that contains object of other classes created.
7. Explain Hybrid inheritance.
8. Illustrate polymorphism means in C++.

9. Differentiate B/W inheritance and polymorphism.

### **10-Mark questions**

#### **Understand**

1. Describe different form of inheritance with Example
2. Explain multiple inheritances. When do we use such an inheritances.
3. Explain public and private member inheritance.
4. Develop some example programs related to classes and objects.
5. Summarize levels of inheritance.

#### **Create**

6. Write a program to demonstrate the multilevel inheritance.
7. C++ program to calculate the area and perimeter of rectangles using concept of inheritance.

### **Unit-6 Pointers, Virtual functions, console I/O statements**

#### **5-Mark questions**

#### **Remember**

1. Describe pointers with syntax.
2. Discuss how procedure makes class virtual.
3. Define Virtual functions.
4. List rules for Virtual functions.
5. List the features of JAVA.
6. List the features of PYTHON.
7. List the features OF C#.

#### **Understand**

8. State Pure virtual functions with example.
9. Discuss the need of Virtual function

#### **10-Mark questions**

#### **Understand**

1. Explain declaration and initialization of pointers with syntax.
2. Explain pointer with arrays with example.
3. Explain how to implement pointers into the objects.
4. Discuss Formatted and Unformatted I/O functions.
5. Describe to make Virtual function "Pure". Explain implications making a function a pure virtual function.
6. Recognize the need of Virtual function.

**End**