

# CS305PC:OBJECT ORIENTED PROGRAMMING USING C++(CS305PC) COURSE PLANNER

## I.Course Overview:

- To understand how C++ improves C with object-oriented features.
- To learn how to write inline functions for efficiency and performance.
- To learn the syntax and semantics of the C++ programming language.
- To learn how to design C++ classes for code reuse.
- To learn how to implement copy constructors and class member functions.
- To understand the concept of data abstraction and encapsulation.
- To learn how to overload functions and operators in C++.
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
- To learn how to use exception handling in C++ programs.

## II.Pre-requisites:

A course on “Programming for Problem Solving using C”.

## III. Course Objectives:

- Introduces Object Oriented Programming concepts using the C++ language.
- Introduces the principles of data abstraction, inheritance and polymorphism
- Introduces the principles of virtual functions and polymorphism
- Introduces handling formatted I/O and unformatted I/O
- Introduces exception handling

## IV.Course Outcomes:

- Able to develop programs with reusability.
- Develop programs for file handling.
- Handle exceptions in programming.
- Develop applications for a range of problems using object-oriented programming techniques.

## V.How program outcomes are assessed:

Program Outcomes (PO)		Level	Proficiency assessed by
PO1	<b>Engineering knowledge:</b> Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems related to Computer Science and Engineering.	2.5	Mini Projects
PO2	<b>Problem analysis:</b> Identify, formulate, review research literature, and analyze complex engineering problems related to Computer Science and Engineering and reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	1.	Lectures, Assignments, Exams
PO3	<b>Design/development of solutions:</b> Design solutions for complex engineering problems related to Computer Science and Engineering and design system components or processes that meet the specified	2.5	Mini Projects

Program Outcomes (PO)		Level	Proficiency assessed by
	needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.		
PO4	<b>Conduct investigations of complex problems:</b> Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	2.5	--
PO5	<b>Modern tool usage:</b> Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	2	--
PO6	<b>The engineer and society:</b> Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the Computer Science and Engineering professional engineering practice.	-	--
PO7	<b>Environment and sustainability:</b> Understand the impact of the Computer Science and Engineering professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	-	Lectures, Assignments, Exams
PO8	<b>Ethics:</b> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	-	
PO9	<b>Individual and team work:</b> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	-	Mini Projects
PO10	<b>Communication:</b> Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	-	--
PO11	<b>Project management and finance:</b> Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.		Lectures, Assignments, Exams
PO12	<b>Life-long learning:</b> Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.		Lectures, Assignments, Exams

#### VI. How program specific outcomes are assessed:

Program Specific Outcomes (PSO)		Level	Proficiency assessed by
PSO1	<b>Foundation of mathematical concepts:</b> To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.	-	Mini Project
PSO2	<b>Foundation of Computer System:</b> The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.	2	Lectures, Assignments, Exams
PSO3	<b>Foundations of Software development:</b> The ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process. Familiarity and	2	Mini Project

	practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research.		
--	---	--	--

## **VII. Course Content:**

### **UNIT - I**

**Object-Oriented Thinking:** Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts Abstraction, Encapsulation, Inheritance and Polymorphism. C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statement- if, switch, while, for, do, break, continue, goto statements. Functions - Scope of variables, Parameter passing, Default arguments, inline functions, Recursive functions, Pointers to functions. Dynamic memory allocation and de-allocation operators-new and delete, Preprocessor directives.

### **UNIT - II**

**C++ Classes and Data Abstraction:** Class definition, Class structure, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction, ADT and information hiding.

### **UNIT - III**

**Inheritance:** Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class. Virtual Functions and Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.

### **UNIT - IV**

**C++ I/O:** I/O using C functions, Stream classes hierarchy, Stream I/O, File streams and String streams, Overloading operators, Error handling during file operations, Formatted I/O.

### **UNIT - V**

**Exception Handling:** Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

### VIII. Lesson plan:

S.NO	WEEK	TOPICS	Course Learning Outcomes	Teaching methodologies	REFERENCES
<b>UNIT-1</b>					
1	I	Introduction to object oriented programming	<b>Understand:</b> object oriented programming	Chalk and board, PPT presentation	T1
2		Different paradigm for problem solving ,need for oops paradigm	<b>Describe: problem solving need</b>		T1
3		Difference between oops and procedure oriented programming	<b>Define: oops and procedure oriented program</b>		T1
4		Overview of oops concepts- encapsulation,abstraction,i nheritance ,polymorphism	<b>Understand: oops concepts</b>		T1
5	2	Introduction to c++,structure of c++ program	<b>Understand: c++ structure</b>		T1
6		Data types, Declaration of variables,	<b>Define: Components and Categories</b>		T1
7		Expressions, Operators, Operator Precedence	<b>Understand: Expression,operators</b>		T1
8		Evaluation of expressions, Type conversions	<b>Evaluate: expressions and type conversion</b>		T1
9	3	Pointers, Arrays, Pointers and Arrays	<b>Understand: Pointers,Arrays,Pointers</b>		T1
10		Strings, Structures, References	<b>Understand: String ,Structures,references</b>		T1
11		Flow control statement- if, switch, while, for, do, break, continue, goto statements	<b>Describe: flow control statements</b>		T1
12		Functions - Scope of variables, Parameter passing, Default arguments, inline functions	<b>Describe: ISDN Physical layer: Transmission modes</b>		T1
13	4	Recursive functions, Pointers to functions.	<b>Describe: functions</b>		T1
14		Pointers to functions.	<b>Understand: Pointer to functions</b>		T1
15		Dynamic memory allocation and de-allocation operators-new and delete.	<b>Understand: Dynamic memory allocation and deallocation</b>		T1
16		Preprocessor directives	<b>Understand: Preprocessor directives</b>		T1
17	5	REVISION			T1

18		MOCK TEST-1			T1
19		<i>Tutorial/bridge class #1</i>			
<b>UNIT-2</b>					
20	5	C++ Classes and Data Abstraction-Introduction	<b>Understand:Classes and data Abstraction</b>	Chalk and board, PPT presentation	T1
21	6	Class definition, Class structure, Class objects	<b>Understand:</b> Class,objects		T1
22		Class scope, this pointer	<b>Describe:</b> Class scope, this pointer		T1
23		Friends to a class	<b>Define:</b> Friends to class		T1
24		Static class members,	<b>Understand:</b> Static class members		T1
25		Constant member functions	<b>Describe:</b> Constant member functions		T1
26	7	Constructors	<b>Understand: constructors</b>		T1
27		Destructors	<b>Understand:</b> destructors		T1
28		Dynamic creation and destruction of objects	<b>Understand:</b> Dynamic creation and destruction of objects		T1
29		Data abstraction	<b>Describe:</b> Data abstraction		T1
30	8	ADT and information hiding	<b>Describe:</b> Information hiding		T1
31		Example Programs	Apply:concepts in writing program.		T1
32		<i>Tutorial/bridge class #2</i>			T1
<b>UNIT-3</b>					
33	9	Inheritance: Defining a class hierarchy	<b>Understand: inheritance</b>	Chalk and board, PPT presentation	T1
34		Different forms of inheritance, Defining the Base and Derived classes	<b>Describe:</b> Different types of inheritance		T1
35		Access to the base class members,	<b>Describe:</b> Access to the base class		T1
36		Base and Derived class construction	<b>Understand:</b> Base and derived class		T1
37	10	Virtual base class.	<b>Understand:</b> Virtual base class		T1
38		Static and Dynamic binding,	<b>Describe:</b> Static and Dynamic binding		T1
39		virtual functions, Dynamic binding through virtual functions	<b>Describe:</b> Virtual functions		T1
40		Virtual function call mechanism, Pure virtual	<b>Describe Pure virtual functions</b>		T1

		functions,				
41	11	classes, Implications of polymorphic use of classes	<b>Describe:</b> implications of polymorphic use of classes		T1	
42		<i>Tutorial/bridge class #3</i>			T1	
<b>UNIT-4</b>						
43	11	C++ I/O: I/O using C functions	<b>Understand: I/O using c functions</b>	Chalk and board, PPT presentation	T1	
44		Stream classes hierarchy	<b>Describe:</b> stream class hierarchy		T1	
45	12	Stream I/O, File streams	<b>Describe:</b> Stream I/O File streams		T1	
46		String streams	<b>Understand:</b> String streams		T1	
47		Overloading operators	<b>Understand:</b> overloading operators		T1	
48		Error handling during file operations	<b>Describe:</b> Error handling		T1	
49	13	Formatted I/O.	<b>Describe:</b> formatted I/O		T1	
50		Example Programs	<b>Apply: writing programs</b>		T1	
51		<i>Tutorial/bridge class #4</i>				
52		<i>Tutorial/bridge class #5</i>				
<b>UNIT-5</b>						
53	14	Exception Handling: Introduction	<b>Describe: Exception handling</b>		Chalk and board, PPT presentation	T1
54		Benefits of exception handling	<b>Discus:</b> Benefits of exception handling	T1		
55		Throwing an exception	<b>Describe:</b> Throwing an exception	T1		
56		The try block	<b>Explain: try block</b>	T1		
57	15	<b>MOCK TEST-2</b>	<b>MOCK TEST-2</b>	T1		
58		Catching an exception	Understand: catching an exception	T1		
59		Exception objects	Understand: Exception objects	T1		
60		Exception specifications	Understand: Exception specifications	T1		
61	16	Stack unwinding	Define : Stack unwinding	T1		
62		Rethrowing an exception	Understand :Rethrowing an exception	T1		
63		Catching all exceptions.	<b>Understand: Catchingall exceptions</b>	T1		
64		<i>Tutorial/bridge class #6</i>				

## II MID EXAMINATIONS (WEEK 17)

### TEXT BOOKS:

1. The Complete Reference C++, 4th Edition, Herbert Schildt, Tata McGraw Hill.
2. Problem solving with C++: The Object of Programming, 4th Edition, Walter Savitch, Pearson Education.

### REFERENCES:

1. The C++ Programming Language, 3rd Edition, B. Stroutstrup, Pearson Education.
2. OOP in C++, 3rd Edition, T. Gaddis, J. Walters and G. Muganda, Wiley Dream Tech Press.
3. Object Oriented Programming in C++, 3rd Edition, R. Lafore, Galigotia Publications Pvt Ltd.

### XI.MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcomes	Program Outcomes (PO)												Program Specific Outcomes (PSO)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	3	1	3	3	2								-	2	2
<b>CO2</b>	3	1	3	3	2								-	2	2
<b>CO3</b>	2	1	2	2	2								-	2	2
<b>CO4</b>	2	1	2	2	2								-	2	2
<b>CO5</b>	2.5	1	2.5	2.5	2									2	2
<b>AVG</b>	3	1	3	3	2								-	2	2

**1: Slight**  
(Low)

**2: Moderate**  
(Medium)

**3: Substantial**  
(High)

**- : None**

# DESCRIPTIVE QUESTIONS

## UNIT-1

### Short Answer Questions

QUESTIONS	Blooms taxonomy level	Course outcomes
1. What are different object oriented Paradigm?	Knowledge	1
2. What is difference between Object oriented and project oriented programming?	Knowledge	1
3. Define abstraction,encapsulation,inheritance and polymorphism?	Knowledge	1
4. What are applications of OOPS?	Knowledge	1
5. What are different I/O operators we use in C++?	Knowledge	1
6. Why we use iostream.h and Namespace in C++?	Knowledge	1
7. What is inline function?	Knowledge	1
8. Why we use new and delete operator?	Knowledge	1

### Long Answer Questions

1. Explain the different properties of object oriented programming?	Understand	1
2. Explain the structure of C++ program with an example?	Understand	1
3. Explain the conditional statements in C++ with an examples?	Understand	1
4. Explain the looping Statements in C++ with an examples?	Understand	1
5. Discuss about the parameter passing in C++?	Knowledge	1
6. Explain the use of pointers in C++?	Understand	1
7. Explain the memory allocation and de-allocation function in C++?	Understand	1

## UNIT-2

### Short Answer Questions

QUESTIONS	Blooms taxonomy level	Course outcomes
1. Define Class and object	Knowledge	2
2. What is a friend function?	Knowledge	2
3. What do you mean by static class?	Knowledge	2
4. What do you mean by constructor and destructor?	Knowledge	2
5. What is data abstraction?	Knowledge	2
6. What is data hiding?	Knowledge	2

## Long Answer Questions

1. Explain the structure of class and object in C++ with a example?	Understand	2
2. Explain the concept of use of constructor and destructor in C++ with an example?	Understand	2
3. Explain the concept of friend function and its use in C++?	Understand	2
4. Explain the concept of Data Abstraction and information hiding?	Understand	2
5. Explain the different access specifier which we use in C++?	Understand	2

## UNIT-3

### Short Answer Questions

<b>QUESTIONS</b>	<b>Blooms taxonomy level</b>	<b>Course outcomes</b>
1. What do you mean by inheritance?	Knowledge	3
2. What are the different types of inheritance?	Knowledge	3
3. What do you mean by virtual class?	Knowledge	3
4. Discuss about static and dynamic binding?	Understand	3
5. What is virtual function?	Knowledge	3
6. What is polymorphism?	Knowledge	3
7. What is abstract class?	Knowledge	3
8. What is virtual destructor?	Knowledge	3

## Long Answer Questions

1. Explain the concept of inheritance with an example?	Understand	3
2. Explain the different forms of inheritance in c++?	Understand	3
3. Explain the concept of virtual function in C++?	Understand	3
4. Explain the concept of Abstract Class in C++?	Understand	3
5. Explain operator overloading and function overloading in C++?	Understand	3
6. Explain the dynamic binding through virtual functions	Understand	3

## UNIT-4

### Short Answer Questions

QUESTIONS	Blooms taxonomy level	Course outcomes
1. What do you understand by stream classes?	Knowledge	4
2. State the different Unformatted input/output Operations?	Knowledge	4
3. State Formatted Console I/O Operations?	Knowledge	4
4. Why we use getline() and write() functions?	Knowledge	4
5. What is the role of iomanip file?	Knowledge	4
6. What are the different steps of file operations?	Knowledge	4
7. What do you mean by error handling during file operations?	Knowledge	4
8. What are input and output streams for files operation?	Knowledge	4

### Long Answer Questions

1. What is a file mode ?describe the various file mode options available.	Knowledge	4
2. Describe briefly the features of I/O system supported by C++.	Understand	4
3. What is the basic difference between manipulators and ios member functions in implementation? Give examples.	Understand	4
4. Explain the different sets for Stream classes for console operations?	Understand	4
5. How is cout able to display various types of data without any special instructions?	Understand	4
6. Explain the different sets of file stream classes	Understand	4

## UNIT-5

QUESTIONS	Blooms taxonomy level	Course outcomes
1. What do you mean by exception handling?	Knowledge	5
2. Describe the role of keywords try,throw and catch in exception handling?	Understand	5
3. When should a program throw an exception?	Knowledge	5

4. When do we used multiple catch handlers?	Understand	5
5.What do you mean by rethrowing an exception	Understand	5

### Short Answer Questions

### Long Answer Questions

1. Explain mechanism of exception handling.	Understand	5
2. What is an exception specification? When is it used?	Knowledge	5
3. Explain in detail about rethrowing an exception?	Understand	5
4. What do you mean by stack unwinding?	Knowledge	5

## UNIT-1

### OBJECTIVE QUESTIONS:

- 1.) \_\_\_\_are the basic run time entities in an object-oriented system.  
A) Class b) **object** c) data d) none
- 2.) The wrapping up of data and function into a single unit (called class) is known as  
a) Data hiding b) **encapsulation** c) insulation d) information
- 3.) \_\_\_\_is the process by which objects of one class acquired the properties of objects of another classes.  
a) **Inheritance** b) classification c) reusability d) all the above
- 4.) The process of making an operator to exhibit different behaviors in different instances is known as \_\_\_\_  
a) information hiding b) function overloading c) **operator overloading** d) none
- 5.) \_\_\_\_is the namespace where ANSI C++ standard class libraries are defined.  
a) keyword b) **std** c) keyword d)directives e) none

### Fill in the blanks:

- 6) \_\_\_\_\_means that the code associated with a given procedure is not known until the time of the run time.
- 7) \_\_\_\_\_involves specifying the name of the object, the name of the function and the information to be sent.
- 8) An \_\_\_\_\_ function is expanded in the line where it is invoked.
- 9) The variables declared within the body of the block are called \_\_\_\_\_and can be used only within the block.
- 10) \_\_\_\_\_means one name, multiple forms. It allows us to have more than one function with the same name in a program.

## UNIT-2

### OBJECTIVE QUESTIONS:

- 1) \_\_\_\_\_ (having the same name as that of the class) is a member function which is automatically used to initialize the objects of the class type with legal initial values  
a) **Constructor** b) destructor c) virtual function d) none
- 2) The operator::known as \_\_\_\_\_  
a) **Access specifier** b) scope resolution c) new d) none
- 3) \_\_\_\_\_ refers to putting together essential features without including background details  
a) Information hiding b) **Data Abstraction** c) Dynamic binding d) None
- 4) The data declared under \_\_\_\_\_ section are hidden and safe from accidental manipulation.  
a) Public b) Protected c) **Private** d) none
- 5) It is of the form classname (classname &) and used for the initialization of an object form another object of same type  
a) **Copy constructor** b) initialization c) overloaded constructor d)none

### Fill in the blanks:

- 6) \_\_\_\_\_ type (not even void) cannot be specified for constructors.
- 7) An object of a class with a constructor cannot be used as a member of a \_\_\_\_\_
- 8) The make implicit calls to the memory allocation and deallocation operators \_\_\_\_\_ and \_\_\_\_\_ are used
- 9) \_\_\_\_\_ is implemented as functions and can be member functions or global functions.
- 10) In C++, the keywords \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ are called access specifiers.

## UNIT-3

### OBJECTIVE QUESTIONS:

- 1) \_\_\_\_\_ is a variable which holds a memory address  
a) Local variable c) global variable c) **pointer** 4) none
- 2) \_\_\_\_\_ is a mechanism of reusing and extending existing classes without modifying them  
a) **Inheritance** b) polymorphism c) reusability d) virtual function
- 3) \_\_\_\_\_ is a mechanism that enables same interface functions to work with the whole class hierarchy  
a) Inheritance b) **polymorphism** c) reusability d) virtual function
- 4) \_\_\_\_\_ also called late binding  
a) Static binding b) **dynamic binding** c) virtual function d)none
- 5) \_\_\_\_\_ is automatically passed to a member function when it is called.  
a) **This pointer** b) friend function c) virtual function d)none

### Fill in the blanks:

- 6) \_\_\_\_\_ inheritance derive from a single class.
- 7) Polymorphism mechanism is supported in C++ by the use of \_\_\_\_\_.
- 8) \_\_\_\_\_ operator is used to deallocate memory
- 9) \_\_\_\_\_ is also called early binding.
- 10) Two types of polymorphism are \_\_\_\_\_ and \_\_\_\_\_

## UNIT-4

### Fill in the blanks:

- 1) A \_\_\_\_\_ is a sequence of bytes and serves as a source or destination for an I/O data.
- 2) The \_\_\_\_\_ operator is overloaded in the istream class as an extraction operator
- 3) The classes istream and ostream define two member functions \_\_\_\_\_ and \_\_\_\_\_
- 4) \_\_\_\_\_ for ifstream functions meaning open for reading only.
- 5) The \_\_\_\_\_ member function closes the file.

### OBJECTIVE QUESTIONS:

- 6) The \_\_\_\_\_ functions shifts the associated file's input file pointer and output file pointer.  
a) **seekg** () b) put() c) get() d) write()
- 7) ios::out for ofstream functions meaning \_\_\_\_\_ for writing only.  
a) **open** b) close c) append() d) none
- 8) \_\_\_\_\_ Provides support for simultaneous input and output operations  
a) ifstream b) ofstream c) **fstream** d) filebuf
- 9) The header file \_\_\_\_\_ provides a set of functions called manipulators which can be used to manipulate the output format  
a) **omanip** b) iostream c) fill() d) none
- 10) The class \_\_\_\_\_ (through inheritance) provides the facilities for formatted output  
a) istream b) **ostream** c) fstream d) none

## UNIT-5

### Fill in the blanks:

- 1) The \_\_\_\_\_ block receives the exception send by the throw block in the try block.
- 2) A \_\_\_\_\_ may decide to rethrow an exception caught without processing them
- 3) Exceptions are basically of two types namely, \_\_\_\_\_ and \_\_\_\_\_
- 4) Exception handling mechanism is basically built upon three keywords \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_
- 5) The \_\_\_\_\_ statement catches an exception whose type matches with the type of catch argument.

### OBJECTIVE QUESTIONS:

- 6) Which type of program is recommended to include in try block?  
a) static memory allocation b) dynamic memory allocation c) const reference d) pointer
- 7) Which statement is used to catch all types of exceptions?  
a) **catch()** b) catch(Test t) c) catch(...) d) none of the mentioned
- 8) What kind of exceptions are available in c++?  
a) **handled** b) unhandled c) static d) dynamic
- 9) Which are the two blocks that are used to check error and handle the error?  
a) **Try and catch** b) Trying and catching c) Do and while d) TryDo and Check
- 10) How many catch blocks can a single try block can have?  
a) Only 1 b) Only 2 c) Maximum 127 d) **As many as required**

