



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

OBJECT ORIENTED PROGRAMMING TUTORIAL QUESTION BANK

1	Department	COMPUTER SCIENCE AND ENGINEERING
2	Course code	ACSD01
3	Course Title	OBJECT ORIENTED PROGRAMMING
4	Class / Semester	I / I
5	Regulation	BT-23
6	Course Coordinator	Dr. J Sirisha Devi

COURSE OBJECTIVES:

The students will try to learn:

I	The fundamental concepts and principles of object-oriented programming in high-level programming languages.
II	Advanced concepts for developing well-structured and efficient programs that involve complex data structures, numerical computations, or domain-specific operations.
III	The design and implementation of features such as inheritance, polymorphism, and encapsulation for tackling complex problems and creating well-organized, modular, and maintainable code.
IV	The usage of input/output interfaces to transmit and receive data to solve real-time computing problems.

COURSE OUTCOMES:

After successful completion of the course, students should be able to:

CO 1	Interpret the features of object-oriented programming languages, comparison, and evolution of programming languages.
CO 2	Model the real-world scenario using class diagrams and exhibit communication between objects.
CO 3	Estimate the need for special functions for data initialization.
CO 4	Outline the features of object-oriented programming for binding the attributes and behavior of a real-world entity.
CO 5	Use the concepts of streams and files that enable data management to enhance programming skills.

CO 6	Develop contemporary solutions to software design problems using object-oriented principles.
------	---

QUESTION BANK:

Q.No	QUESTION	Taxonomy	How does this subsume the level	CO's
MODULE I				
OBJECT-ORIENTED CONCEPTS				
PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				
1	What is the need for object oriented programming?	Understand	Recall the concept of object oriented programming to justify the need for OOP.	CO 1
2	What are the limitations of OOP?	Remember		CO 1
3	Where do we use OOPS in real life?	Understand	Recall the concept of object oriented programming to justify the use of OOP.	CO 1
4	What is top down process of reading?	Remember		CO 1
5	What are the benefits of top-down processing?	Understand	Recall the concept of programming and summarize the benefits of top-down processing.	CO 1
6	How do we use top-down processing everyday?	Understand	Recall the concept of top-down processing to mention its usage in everyday life.	CO 1
7	What is the bottom up approach in language teaching?	Understand	Recall the concept of programming and mention the uses of bottom up approach in teaching.	CO 1
8	What is an example of bottom-up learning?	Understand	Recall the concept of bottom-up processing to give few examples.	CO 1
9	Define abstraction in software in general.	Remember		CO 1

10	How to achieve data abstraction in OOP?	Understand	Recall the concept of data abstraction to narrate the process of data abstraction.	CO 1
PART-B LONG ANSWER QUESTIONS				
1	Discuss about the objects and legacy systems.	Understand	Recall the concept of object oriented programming to explain objects and legacy systems.	CO 1
2	Differentiate between procedure-oriented programming and object-oriented programming.	Understand	Recall the concepts of procedure and oriented programming and object oriented programming.	CO 1
3	Explain the features of Object-oriented programming.	Understand	Recall the concept of OOP and explain its features.	CO 1
4	What are the applications of Object-oriented programming?	Understand	Recall the concept of OOP and explain its features.	CO 1
5	Explain top-down and bottom-up approaches of problem solving.	Remember	Recall the concepts of top down and bottom up approaches.	CO 1
6	Explain the mechanism of data abstraction.	Remember		CO 1
7	What are the advantages and disadvantages of Object-oriented programming?	Understand	Recall the concepts of object oriented programming to explain its advantage and disadvantages.	CO 1
8	Differentiate between top-down approaches and bottom-up approaches.	Understand	Recall the concepts of top down and bottom up approaches to differentiate them.	CO 1
9	How object oriented programming differs from object based programming language? Discuss the benefits of OOP.	Understand	Recall the concepts of objects to understand benefits of OOP	CO 1
10	Explain how procedural abstraction enhances code readability and reusability.	Understand	Recall the concept of data abstraction to enhance code readability and reusability.	CO 1

11	Explore the concept of encapsulation as an abstraction mechanism.	Remember		CO 1
12	How does encapsulation facilitate information hiding and data protection in object-oriented programming?	Understand	Recall the concept of encapsulation to explain about information hiding and data protection.	CO 1
13	Explain any two features of object-oriented programming with an example.	Understand	Recall the features of object oriented programming.	CO 1
14	Discuss the advantages and disadvantages of top-down and bottom-up approaches.	Understand	Recall the concepts of top down and bottom up approaches to explain their advantages and disadvantages.	CO 1
15	Explain the various applications of Object Oriented Programming using an example.	Understand	Recall the concept of object oriented programming to explain its applications.	CO 1
16	Illustrate the benefits of Object-Oriented Programming.	Understand	Recall the concepts of objects to understand benefits of OOP	CO 1
17	What is the purpose of the object-oriented programming and its need in the current industry?	Understand	Recall the concept of object oriented programming to justify its need in the current industry.	CO 1
18	Define abstraction. What are the different layers of abstraction?	Remember		CO 1
19	Explain the various forms of abstraction.	Remember		CO 1
20	Discuss the advantages and challenges of designing software with a hierarchical structure of abstraction layers.	Understand	Recall the concept of data abstraction to explain its challenges in designing software.	CO 1
PART-C SHORT ANSWER QUESTIONS				
1	What is a legacy system?	Remember		CO 1

2	What are the characteristics of legacy systems? Name a few legacy systems.	Understand	Recall the characteristics of legacy systems	CO 1
3	Define procedure-oriented programming (POP).	Remember		CO 1
4	Define object-oriented programming (OOP).	Remember		CO 1
5	What are the four cornerstones of OOP?	Remember		CO 1
6	What is platform independency?	Remember		CO 1
7	Define abstraction.	Remember		CO 1
8	Define datatype and its types.	Remember		CO 1
9	List the 8 primitive datatypes.	Remember		CO 1
10	What are syntax rules?	Understand	Recall the concept of programming to explain syntax rules.	CO 1
11	Define data hiding.	Remember		CO 1
12	Define object data.	Remember		CO 1
13	What is object behaviors?	Remember		CO 1
14	Define method.	Remember		CO 1
15	Define attribute.	Remember		CO 1
16	Define interface.	Remember		CO 1
17	Define data abstraction.	Remember		CO 1
18	Define process abstraction.	Remember		CO 1
19	What is information hiding?	Remember		CO 1
20	What is a message?	Remember		CO 1
MODULE II				
CLASSES AND OBJECTS				
PART-A PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				
1	How much memory does a class occupy?	Understand	Recall the concept of class and memory allocation.	CO 2

2	is it always necessary to create objects from class?	Understand	Recall the concept of classes and objects.	CO 2
3	Are class and structure the same? If not, what is the difference between a class and a structure?	Understand	Recall the concept of class and structure	CO 2
4	Can a class extend itself?	Understand	Recall the concept of classes and its properties.	CO 2
5	What is the difference between a class and object?	Understand	Recall the concept of classes and objects.	CO 2
6	What is the need of access specifiers?	Understand	Recall the concept of access specifiers.	CO 2
7	What will happen if a class extends two interfaces and they both have a method with same name and signature?	Understand	Recall the concept of classes and its properties.	CO 2
8	Which access specifier should be used in a class where the instances can't be created?	Understand	Recall the concept of access specifiers and types of access specifiers.	CO 2
9	On which specifier's data, does the size of a class's object depend?	Understand	Recall the concept of classes and access specifiers.	CO 2
10	What is the difference between local variable and data member?	Understand	Recall the concepts of members of a class and local variables.	CO 2
PART-B LONG ANSWER QUESTIONS				
1	Define class and object with suitable example. How members of class can be accessed?	Understand	Recall the concepts of classes, objects, member variables and memeber functions.	CO 2
2	What is a method? How a method is defined?	Understand	Recall the concept of methods.	CO 2
3	What are access specifiers and what is their significance in OOP?	Understand	Recall the concept of access specifiers, modes of access specifiers.	CO 2

4	What is the difference between public, private and protected access modifiers?	Understand	Recall the concept of access specifiers, modes of access specifiers.	CO 2
5	Create a real scenario where static data members are useful. Explain with suitable example.	Apply	Recall the concept of static data members and its usage for problem solving.	CO 2
6	Explain the scope of access specifiers.	Understand	Recall the concept of access specifiers, modes of access specifiers.	CO 2
7	Draw a class diagram for hospital management system which consists of classes, their attributes, operations (or methods), and relationship among objects. The main classes are hospitals, patients, doctors, nurses, appointments, and medicines.	Apply	Recall the concepts of classes and class diagram.	CO 2
8	Draw a class diagram for Library Management System. This diagram must describe the structure of the system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects	Apply	Recall the concepts of classes and class diagram.	CO 2
9	What is the major use of data members and member functions as static? Explain it with an example.	Apply	Recall the concepts of static data members and static member functions.	CO 2
10	In object-oriented programming, what is the difference between accessing a class's public member and a private member?	Understand	Recall the concepts of private members and public members and their scope of accessibility.	CO 2
11	Explain object behaviors with examples.	Understand	Recall the concept of objects and their behavior.	CO 2

12	Explain the basic elements of a class diagram with an example.	Analyze	Recall the concepts of classes and class diagrams.	CO 2
13	Why static members are used in a class, and what advantages do they offer in terms of memory and performance?	Understand	Recall the concepts of static data members, static member functions, and memory allocation.	CO 2
14	Explain the difference between a static method and an instance method in terms of access and usage.	Understand	Recall the concepts of accessing static methods.	CO 2
15	In object-oriented programming, what is the significance of the static keyword when applied to members and methods?	Understand	Recall the concepts of static data members and static member functions.	CO 2
16	Give an example of a real-world problem where static members and methods are useful for solving the problem efficiently.	Apply	Recall the concepts of static data members and static member functions.	CO 2
17	What are some potential drawbacks or limitations of using static members and methods, and when should you be cautious about their use?	Understand	Recall the concepts of static data members, and static member functions.	CO 2
18	What is memory allocation, and why is it important in computer systems?	Understand	Recall the concept of static and dynamic memory allocation.	CO 2
19	Describe the concept of dynamic memory allocation and its advantages in program flexibility.	Understand	Recall the concept of dynamic memory allocation.	CO 2
20	What is virtual memory, and how does it extend physical memory allocation to improve system performance?	Understand	Recall the concept of virtual memory.	CO 2

PART-C SHORT ANSWER QUESTIONS				
1	What is a class?	Remember		CO 2
2	What is an object?	Remember		CO 2
3	What is an interface?	Remember		CO 2
4	What is method hiding?	Remember		CO 2
5	How many types of access specifiers are provided in OOP?	Understand	Recall the concept of access specifiers.	CO 2
6	What is an instance of a class?	Remember		CO 2
7	Define class diagram.	Remember		CO 2
8	List out the vital components of a class diagram.	Understand	Recall the concept of class diagrams.	CO 2
9	What is the purpose of a class diagram?	Understand	Recall the concept of class diagrams.	CO 2
10	What are the benefits of class diagrams?	Understand	Recall the concept of class diagrams.	CO 2
11	Define public access specifier.	Remember		CO 2
12	Define private access specifier.	Remember		CO 2
13	Define protected access specifier.	Remember		CO 2
14	What is the difference between default and protected access specifier?	Understand	Recall the concept of access specifiers.	CO 2
15	Define aggregation.	Remember		CO 2
16	Define generalization.	Remember		CO 2
17	Define association.	Remember		CO 2
18	What is a static member?	Remember		CO 2
19	What is a static method?	Remember		CO 2
20	What is the need for static members?	Understand	Recall the concept of static members.	CO 2
MODULE III				

SPECIAL MEMBER FUNCTIONS AND OVERLOADING				
PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				
1	Can we call sub class constructor from super class constructor?	Understand	Recall the concept of constructors.	CO 3
2	What happens if you keep a return type for a constructor?	Understand	Recall the concept of constructors.	CO 3
3	When do we need constructor overloading?	Understand	Recall the concept of constructors and overloading.	CO 3
4	How a no – argument constructor is different from default Constructor	Understand	Recall the concept of constructors.	CO 3
5	“An overloaded function appears to perform different activities depending the kind of data send to it” Justify the statement with appropriate example.	Understand	Recall the concept of overlaoding.	CO 3
6	If class B inherits class A privately. And class B has a friend function. Will the friend function be able to access the private member of class A?	Understand	Recall the concept of friend functions and inheritance.	CO 3
7	What happens if you do not provide any constructor to a class?	Understand	Recall the concept of constructors.	CO 3
8	If a class has all the private members, which specifier will be used for its implicit constructor?	Understand	Recall the concept of constructors.	CO 3
9	How many friend functions can a class have?	Understand	Recall the concept of friend functions.	CO 3
10	Does friend function violate encapsulation?	Understand	Recall the concept of friend function and encapsulation.	CO 3
PART-B LONG ANSWER QUESTIONS				

1	What is constructor? Explain the concept of default and default copy with suitable example.	Understand	Recall the concept of constructors and types of constructors.	CO 3
2	How are constructors different from member functions? Explain with an example.	Analyze	Recall the concept of constructors and member functions to explain the difference between them with an example.	CO 3
3	Depict the difference between private and public derivation. Explain derived class constructor with suitable example.	Understand	Recall the concept of constructors and types of constructors.	CO 3
4	Explain the default action of the copy constructor. Give a suitable example that demonstrates the technique of overloading the copy constructor.	Understand	Recall the concepts of copy constructor and overloading.	CO 3
5	Write short notes on: a) Early binding, b) late binding.	Understand	Recall the concept of data binding.	CO 3
6	Explain types of constructor with example.	Understand	Recall the concept of constructors and types of constructors.	CO 3
7	What is copy constructor? When it is used implicitly for what purpose?	Understand	Recall the concept of copy constructor.	CO 3
8	What is the difference between copy constructor and default constructor?	Understand	Recall the concept of constructors and types of constructors.	CO 3
9	Describe unary operator overloading along with syntax and example	Apply	Recall the concept of operator overloading.	CO 3
10	Describe binary operator overloading along with syntax and example	Apply	Recall the concept of operator overloading.	CO 3

11	What is overloading of an operator? When it is necessary to overload an operator?	Understand	Recall the concept of operator overloading.	CO 3
12	Differentiate between the concepts of overloading and overriding with an example.	Understand	Recall the concept of overloading and overriding to differentiate between them.	CO 3
13	What is the difference between constructor overloading and operator overloading?	Understand	Recall the concept of overloading.	CO 3
14	Explain is constructor overloading with an example.	Understand	Recall the concepts of copy constructor and overloading.	CO 3
15	Define static data members and explain with an example.	Understand	Recall the concept of static members.	CO 3
16	What is the use of operator overloading? How to overload post and pre-increment operators?	Analyze	Recall the concept of operator overloading.	CO 3
17	Differentiate between method overloading and method overriding.	Understand	Recall the concept of overloading and overriding to differentiate between them.	CO 3
18	What is the concept of friend function? How it violates the data hiding principle? Justify with example.	Understand	Recall the concept of friend functions.	CO 3
19	List out the guidelines that should be followed while using friend functions.	Understand	Recall the concept of friend functions.	CO 3
20	What are friend classes? Explain the advantages of using friend classes.	Understand	Recall the concept of friend classes.	CO 3
PART-C SHORT ANSWER QUESTIONS				
1	Define a constructor.	Remember		CO 3
2	Define constructor chaining.	Remember		CO 3

3	What is No-arg constructor?	Understand	Recall the concept of constructors.	CO 3
4	Define a copy constructor?	Remember		CO 3
5	What are distinguishing characteristics of copy constructors?	Understand	Recall the concept of copy constructors.	CO 3
6	What is a dynamic constructor?	Understand	Recall the concept of constructors.	CO 3
7	What is dynamic initialization?	Understand	Recall the concept of initialization of variables and objects.	CO 3
8	Define destructor?	Remember		CO 3
9	What is the use of destructors?	Understand	Recall the concept of destructors.	CO 3
10	Define method overloading.	Remember		CO 3
11	Define method overriding.	Remember		CO 3
12	What are the different types of constructors?	Understand	Recall the concept of constructors.	CO 3
13	What are distinguishing characteristics of default constructors?	Understand	Recall the concept of constructors and types of constructors.	CO 3
14	What are distinguishing characteristics of overloaded constructors?	Understand	Recall the concept of constructors and types of constructors.	CO 3
15	What is operator overloading?	Remember		CO 3
16	What is unary operator overloading?	Understand	Recall the concept of operator overloading.	CO 3
17	Name the binary operators that can be overloaded.	Understand	Recall the concept of operator overloading.	CO 3
18	Define friend function.	Remember		CO 3
19	What are the characteristics of friend functions?	Understand	Recall the concept of friend functions.	CO 3
20	What is the difference between friend function and friend class in OOP?	Understand	Recall the concept of friend functions and friend class.	CO 3

MODULE IV				
INHERITANCE AND POLYMORPHISM				
PART A- PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				
1	What are the limitations of inheritance?	Understand	Recall the concept of inheritance.	CO 4
2	How is an abstract class different from an interface?	Understand	Recall the concept of abstract class.	CO 4
3	Why do we need to use inheritance?	Understand	Recall the concept of inheritance.	CO 4
4	How will you prove that the features of superclass are inherited in subclass?	Understand	Recall the concept of inheritance.	CO 4
5	Will a constructor and instance initialization blocks inherited to subclass?	Understand	Recall the concepts of inheritance and constructors.	CO 4
6	Are static members inherited to subclass?	Understand	Recall the concept of static members.	CO 4
7	What is the order of calling constructors in case of inheritance?	Understand	Recall the concepts of inheritance and constructors.	CO 4
8	How will you restrict a member of a class from inheriting its subclass?	Understand	Recall the concept of inheritance.	CO 4
9	What happens if both superclass and subclass have a field with the same name?	Understand	Recall the concept of inheritance.	CO 4
10	Can you call the base class method without creating an instance?	Understand	Recall the concept of inheritance.	CO 4
PART-B LONG ANSWER QUESTIONS				
1	Briefly explain types of inheritance used in object oriented Programming.	Understand	Recall the concept of inheritance in object oriented programming.	CO 4
2	Differentiate between multiple inheritance and multilevel inheritance.	Understand	Recall the concepts of multiple inheritance and multilevel inheritance.	CO 4

3	What is the difference between inheritance and encapsulation?	Understand	Recall the concepts of inheritance and encapsulation.	CO 4
4	Discuss the difference between inheritance and abstraction with real time examples.	Understand	Recall the concepts of inheritance and abstraction.	CO 4
5	What is the difference between polymorphism and inheritance?	Understand	Recall the concepts of inheritance and polymorphism.	CO 4
6	How composition differs from inheritance in OOP? Explain with examples.	Understand	Recall the concept of composition.	CO 4
7	How dynamic binding is achieved in polymorphism?	Understand	Recall the concept of polymorphism.	CO 4
8	What is the difference between "is-a" and "has-a" relationship?	Understand	Recall the basic properties of OOP	CO 4
9	Explain polymorphism briefly. Write down the roles of various types of polymorphism.	Understand	Recall the concept of polymorphism and its types.	CO 4
10	How can we achieve dynamic polymorphism briefly? Explain with example.	Understand	Recall the concept of polymorphism and its types.	CO 4
11	What are the types of Polymorphism?	Understand	Recall the concept of polymorphism and its types.	CO 4
12	Explain Compile-time Polymorphism with example.	Understand	Recall the concept of polymorphism and its types.	CO 4
13	Explain overloading unary and binary operators with suitable examples.	Understand	Recall the concept of overloading.	CO 4
14	What are the differences between 'this' and 'super' keyword?	Understand	Recall the various ways to inherit constructor properties to its subclass.	CO 4
15	What are the rules to be followed while overriding a method?	Understand	Recall the concept of overriding.	CO 4

16	Explain public, private and protected access specifiers and show their visibility when they are inherited as public, private and protected.	Understand	Recall the concept of access specifiers and inheritance.	CO 4
17	What do you mean by constructors in derived classes? If a constructor function is defined for derived and base class, then which constructor function gets executed first, explain with example.	Analyze	Recall the concept of constructors and inheritance.	CO 4
18	What do you mean by hybrid inheritance? Explain how to remove ambiguity in the case of hybrid inheritance?	Understand	Recall the various types of inheritance.	CO 4
19	Explain different visibility modes with example.	Understand	Recall the concepts of inheritance and access specifiers.	CO 4
20	How do the properties of following two derived class differ: i. class D1: private B, public C; ii. class D2: protected B, private C;	Analyze	Recall the concepts of inheritance and access specifiers.	CO 4

PART-C SHORT ANSWER QUESTIONS

1	Define inheritance.	Remember		CO 4
2	Define base class.	Remember		CO 4
3	Define derived class.	Remember		CO 4
4	What is an abstract class?	Remember		CO 4
5	What is IS-A relationship?	Remember		CO 4
6	What is a subclass?	Understand	Recall the concept of inheritance	CO 4
7	Define a superclass?	Understand	Recall the concept of inheritance	CO 4

8	What is hybrid inheritance?	Understand	Recall the concept of inheritance	CO 4
9	What is hierarchical inheritance?	Understand	Recall the concept of inheritance	CO 4
10	What is polymorphism?	Remember		CO 4
11	What is static polymorphism?	Understand	Recall the concept of polymorphism	CO 4
12	What is dynamic polymorphism?	Understand	Recall the concept of polymorphism	CO 4
13	List out the operators which cannot be overloaded.	Understand	Recall the concept of operator overloading	CO 4
14	Define virtual function.	Remember		CO 4
15	What is a pure virtual function?	Understand	Recall the concept of virtual functions.	CO 4
16	Differentiate between a class and a method.	Understand	Recall the concept of class.	CO 4
17	Can we pass an object of a subclass to a method expecting an object of the super class?	Understand	Recall the concept of inheritance.	CO 4
18	What is the purpose of the "super" keyword?	Understand	Recall the concept of constructor and inheritance.	CO 4
19	Define virtual base class.	Remember		CO 4
20	What is "final" keyword for method?	Remember		CO 4

MODULE V

CONSOLE I/O AND WORKING FILES

PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

1	Draw console stream class hierarchy and explain its members.	Understand	Recall the concept of streams.	CO 5
2	What is a stream class for console operations?	Understand	Recall the concept of streams and its operations.	CO 5

3	Compare and contrast batch processing and stream processing. When would you choose one over the other for data processing tasks?	Understand	Recall the concept of data processing.	CO 5
4	Classify the methods of unformatted console input/output operations with example.	Understand	Recall the concept of I/O operations.	CO 5
5	Is it possible to dynamically change the formatting of console output during program execution? How might you achieve this, and what use cases could benefit from such dynamic formatting?	Understand	Recall the concept of I/O operations formatting.	CO 5
6	Are there any challenges or considerations related to using manipulators for console output when targeting multiple operating systems or platforms? How can you ensure consistent formatting across platforms?	Understand	Recall the concepts of manipulators.	CO 5
7	What is a file structure in operating systems?	Understand	Recall the concept of files.	CO 5
8	How does the use of manipulators impact the readability and user-friendliness of console output? Can you provide examples where well-chosen manipulators significantly improve output presentation?	Understand	Recall the concepts of manipulators.	CO 5
9	Explain the primary file operations involved in file handling, such as opening, reading, writing, and appending to files.	Understand	Recall the concept of file operations.	CO 5

10	What is an InputStream? Outline the methods defined by InputStream.	Understand	Recall the concept of streams.	CO 5
PART-B LONG ANSWER QUESTIONS				
1	Write a short note on Character Stream classes.	Understand	Recall the concept of streams and types of streams.	CO 5
2	Write short notes on: a) Manipulators, b) protected access specifier	Understand	Recall the concept of access specifiers and manipulators.	CO 5
3	How is character I/O different from binary I/O? Explain with examples.	Understand	Recall the formats of I/O operations on character and binary files.	CO 5
4	List and explain in brief various functions required for random access file operations.	Understand	Recall the functions to work with random access files.	CO 5
5	What are manipulators? How can you create your own manipulators? Explain with an example.	Understand	Recall the concept of manipulators.	CO 5
6	Explain how files can be processed and data can be manipulated during file I/O operations.	Understand	Recall the concept of manipulators.	CO 5
7	Compare and contrast appending data to a file with overwriting the entire file. When would you choose one approach over the other?	Understand	Recall the operations on files.	CO 5
8	Describe the significance of closing files after completing file I/O operations. What can happen if files are not properly closed?	Understand	Recall the operations on files.	CO 5
9	Discuss the concept of file locking and its importance in multi-threaded or multi-process applications.	Understand	Recall the concept of file locking.	CO 5

10	Discuss the distinction between character streams and byte streams in unformatted I/O. When would you choose one over the other?	Understand	Recall the concept of streams and I/O formatting.	CO 5
11	Describe the concept of stream buffering in unformatted I/O operations. How does buffering impact the performance of I/O operations?	Understand	Recall the concept of streams and I/O formatting.	CO 5
12	Explain the fundamental difference between unformatted I/O and formatted I/O operations in programming. How do these two approaches impact the way data is read from and written to input and output streams?	Understand	Recall the concept of I/O formatting.	CO 5
13	In the context of unformatted I/O, discuss the importance of documentation and code readability.	Understand	Recall the concept of unformatted I/O operations.	CO 5
14	How can streams be used securely, especially when dealing with untrusted data sources? What precautions should be taken to prevent security vulnerabilities related to streams?	Understand	Recall the concept of streams.	CO 5
15	What are some popular libraries or frameworks for stream processing, and how do they simplify the development of stream-based applications?	Understand	Recall the concept of streams and its applications.	CO 5

16	Describe the concept of stream abstraction layers in programming. Provide a hierarchy or diagram that illustrates the relationship between different levels of stream classes.	Understand	Recall the concept of data abstraction and streams.	CO 5
17	Explore the concept of dynamic formatting with manipulators. How can manipulators be applied dynamically during program execution based on user input or other runtime conditions?	Understand	Recall the concept of manipulators.	CO 5
18	Explain how manipulators can be useful for formatting output in command-line applications.	Understand	Recall the concept of command line arguments	CO 5
19	Discuss the potential consequences of failing to manage resources properly. What happens if you forget to close a stream after using it?	Understand	Recall the concept of command line arguments	CO 5
20	Explain the concept of command line arguments and their role in passing information to a program from the command line.	Understand	Recall the concept of command line arguments	CO 5

PART-C SHORT ANSWER QUESTIONS

1	Define stream.	Remember		CO 5
2	What is the function of console?	Understand	Recall the concept of console I/O.	CO 5
3	What are the two types of I/O stream?	Understand	Recall the concept of streams.	CO 5
4	How character streams are defined?	Understand	Recall the concept of streams.	CO 5
5	What is the difference between a file and a stream?	Understand	Recall the concept of streams and files.	CO 5

6	What is the difference between stream and console I/O?	Understand	Recall the concept of console I/O and stream.	CO 5
7	What is an unformatted I/O operations?	Understand	Recall the concept of unformatted I/O operations.	CO 5
8	What are the manipulators in OOP?	Remember		CO 5
9	What is an output manipulator?	Remember		CO 5
10	What is the purpose of manipulators?	Understand	Recall the concept of manipulators.	CO 5
11	What are manipulative methods?	Understand	Recall the concept of manipulators.	CO 5
12	How do you manage manipulators?	Understand	Recall the concept of manipulators.	CO 5
13	What is a file in OOP?	Remember		CO 5
14	What are the file access right?	Understand	Recall the concept of access rights on files.	CO 5
15	What are the basic operations on a file?	Understand	Recall the concept of files and its operations.	CO 5
16	Compare unformatted data and formatted data.	Understand	Recall the concept of unformatted and formatted I/O operations.	CO 5
17	Which method can be used to open a file in file handling?	Understand	Recall the operations on files.	CO 5
18	How many types of files are there in file handling?	Understand	Recall the types of files.	CO 5
19	What is command line argument in OOP?	Remember		CO 5
20	What are the types of command line arguments?	Understand	Recall the concept of command line arguments.	CO 5

Signature of Course Coordinator
Dr. J Sirisha Devi, Associate Professor

HOD CSE