

UNIT I- OVERVIEW

1. Give some characteristics of procedure-oriented language.

The characteristics of procedure-oriented language are,

- i. Emphasis is on doing things (algorithms).
- ii. Larger programs are divided into smaller programs known as functions.
- iii. Most of the functions share global data.
- iv. Data move openly around the system from function to function.
- v. It employs top-down approach in program design.

2. What are the basic concepts of OOPS?

The basic concepts of OOPS are,

- i. Objects.
- ii. Classes.
- iii. Data abstraction and Encapsulation.
- iv. Inheritance.
- v. Polymorphism.
- vi. Dynamic binding.
- vii. Message passing.

3. What is an object?

An object is basic run-time entity in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle. Each object has the data and code to manipulate the data and these objects interact with each other.

4. What is a class?

A class is a collection of objects of similar type. Once a class has been defined, we can create any number of objects belonging to the class. Class is a user-defined data type and behaves like built-in types of the programming language.

5. What is an encapsulation?

Wrapping up of data and function within the structure is called as encapsulation. The insulation of data from direct access by the program is called as data hiding or information binding. The data is not accessible to the outside world and only those functions, which are wrapped in the class, can access it.

6. What is meant by dynamic binding or late binding?

Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at the run-time.

7. Write the process of programming in an object-oriented language?

The process of programming in an object-oriented language are,

- i. Create classes that define objects and their behavior.
- ii. Creating objects from class definition.
- iii. Establishing communication among objects.

8. List any four advantages of OOPS.

The advantages of OOPS are,

- i. The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
- ii. It is possible to have multiple instances of an object to co-exist without any interference.
- iii. Object oriented programming can be easily upgraded from small to large systems.
- iv. Software complexity can be easily managed.

9. What are the features required for object-based programming language?

The features required for object-based programming are,

- i. Data encapsulation.
- ii. Data hiding and access mechanisms.
- iii. Automatic initialization and clear up of objects.
- iv. Operator overloading.

10. Give any four applications of the OOPS.

The four applications of OOPS are,

- i. Real-time systems.
- ii. Simulation and modeling.
- iii. Object-oriented databases.
- iv. AI and expert systems.

11. What are the operators available in C++?

The operators available in C++ are,

- i. :: - Scope resolution operator.
- ii. :: * - Pointer-to-member declarator .

- iii. ->* - Pointer-to-member operator.
- iv. .* - Pointer-to-member operator.
- v. delete - Memory release operator .
- vi. endl - Line feed operator
- vii. new - Memory allocation operator
- viii. setw - Field width operator

12. . What is a default argument?

Default argument assigns a default value to the parameter, which does not have matching argument in the function call. Default values are specified when the function is declared.

Example:

```
float amount (float principle, int period, float rate=0.15)
{
}
```

13. What is constant argument?

Keyword is const. The qualifier const tells the compiler that the function should not modify the argument. The compiler will generate an error when this condition is violated. This type of declaration is significant only when we pass arguments by reference or pointers

Example:

```
int strlen (const char *p);
```

14. How do you create an object?

Once the class has been declared, we can create object for a class using the class name.

Example:

```
classname x; //memory for x is created
```

15. How the class is specified?

Generally class specification has two parts. They are,

- i. Class declaration - It describes the type and scope of its member
- ii. Class function definition - It describes how the class functions are

impleme

nt ed The

general form is:

```
class class_name
{
```

```

private:
    variable
    declaratio
    ns;
    function
    declaratio
    n;
public:
    variable
    declaration; function
    declaration;
};

```

16. How do you access a class member?

We can access the member function by using the following syntax: Syntax:

Object-name. Function-name (actual arguments); Example:

```
x.getdata (100, 75.5);
```

17. How are the member functions defined?

Member functions can be defined in two ways. They are,

- i. Outside the class definition - Member function can be defined by using scope resolution operator (::).

General format is

```

return type class_name:: function-name (argument declaration)
{
    .....//function body
}

```

- ii. Inside the class definition - This method of defining member function is to replace the function declaration by the actual function definition inside the class. It is treated as inline function

Example:

```

class
item
{
    int a, b;
    void getdata (int x, int y)
    {
        a=x;
    }
};

```

18. What are the features of static data member?

The features of static data member are,

- i. It is initialized to zero when the first object is created
- ii. Only one copy of that member is created for the entire class
- iii. It is only visible within the class
- iv. It is stored separately rather than objects

19. What are the situations that inline functions may not work?

The situations that inline functions may not work is,

- i. For function returning values, if a loop, a switch, or a goto exists.
- ii. For function not returning values, if a return statement exists.
- iii. If function contains static variables.
- iv. If inline functions are recursive.
- v.

20. What is a namespace?

The namespaces are used to group the entities like class, variables, objects, function under a name.

21. What is abstract class?

The class that does not specify the implementation details, but specifies only what are the operation in the particular data structure.

UNIT II- BASIC CHARACTERISTICS OF OOP**1. Define Constructor.**

A constructor is defined as a special member of a class, which has no return type. It can be used to initialize an object immediately upon creation.

2. Write some special characteristics of constructor.

Some special characteristics of constructor are,

- a. They should be declared in the public section.
- b. They are invoked automatically when the objects are created.
- c. They do not have return types, not even void and therefore, and they cannot return values.
- d. They cannot be inherited, though a derived class can call the base class
- e. They can have default arguments.
- f. Constructors cannot be virtual function.

3. How the objects are initialized dynamically?

The objects are initialized dynamically to call parameterized constructor or methods we should pass values to the object i.e., for the constructor `integer add(int a, int b)` it is invoked by `integer a (10, 18)`. This value can be get during run time. i.e., for above constructor

Example:

```
int p,q;  
cin>>p>>q;  
integer add(p,q);
```

4. What is meant by operator overloading?

Operator overloading is the mechanism of giving such special meanings to an operator is known. It provides a flexible option for the creation of new definitions for C++ operators.

5. List out the operators that cannot be overloaded.

The operators that cannot be overloaded are,

- a. Class member access operator (`., .*`)
- b. Scope resolution operator (`::`)
- c. Size operator (`sizeof`)
- d. Conditional operator (`? :`)

6. List out the operators that cannot be overload as friend functions.

The operators that cannot be overload as friend function are:

- a. Assignment operator =
- b. Function call operator ()
- c. Array subscript operator []
- d. Access to class member using pointer to object operator ->

7. What is the purpose of using operator function? Write its syntax.

To define an additional task to an operator, we must specify what it means in relation to the class to which the operator is applied. This is done by Operator function , which describes the task. Operator functions are either member functions or friend functions.

The general form is :

```
return type classname :: operator op(arglist )
{
    // function body
}
```

8. Write any four rules for operator overloading.

Rules for Operator overloading are,

- a. Only the existing operators can be overloaded.
- b. The overloaded operator must have atleast one operand that is of user defined data type.
- c. The basic meaning of the operator should not be changed.
- d. Overloaded operators follow the syntax rules of the original operators.
- e. They cannot be overridden.

9. What are the different types of conversion?

The different types of conversions are,

- a. Basic type to Class type.
- b. Class type to Basic type. c.
- Class type to Class type. d.
- Basic type to Basic type

10. Write the syntax to convert from class to basic type.

The syntax to convert from class to basic type is, Operator type_name ()

```

{
    .....
    Function statements;
    .....
}

```

11. What are the conditions to satisfy the type casting function?

The conditions to satisfy the type casting function are,

- a. It must be a class member.
- b. It must not specify a return type.
- c. It must not have any arguments.

12. What is Friend function? Write the syntax.

A function that has access to the private member of the class but is not itself a member of the class is called friend functions. Friend function is preceded by the keyword „friend“.

The general form is:

```
friend datatype function name (object dec);
```

13. Write some properties of friend function.

The properties of friend function are,

- a. Friend function is not in the scope of the class to which it has been declared as friend. Hence it cannot be called using the object of that class.
- b. Usually it has object as arguments.
- c. It can be declared either in the public or private part of a class.
- d. It cannot access member names directly. It has to use an object name and dot membership operator with each member name. eg: (A . x)

14. What is function overloading? Give an example.

Function overloading means we can use the same function name to create functions that perform a variety of different tasks.

Example: An overloaded add () function handles different data types as shown below.

// Declarations

```
int add (int a, int b); //add function with 2 arguments of same type
int add (int a, int b, int c); //add function with 3 arguments of same type
```

15. Write at least four rules for Operator overloading.

Rules for Operator overloading are,

- a. Only the existing operators can be overloaded.
- b. The overloaded operator must have at least one operand that is of user defined data type.
- c. The basic meaning of the operator should not be changed.
- d. Overloaded operators follow the syntax rules of the original operators.
- E. They cannot be overridden.

16. What is meant by static invocation of virtual function?

It is possible to call virtual function using as object of the class or using a scope resolution operator. In that case, virtual function is invoked statically. This is known as static invocation of the virtual function. The syntax for static invocation of virtual function is,

→
pointer-object classname::virtual-function();

17. What is meant by inheritance?

Inheritance is the process by which objects of one class acquire the properties of another class. It supports the concept of hierarchical classification. It provides the idea of reusability. We can add additional features to an existing class without modifying it by deriving a new class from it.

18. What is meant by visibility mode? Mention the visibility modes available.

Visibility mode specifies whether the features of the base class are privately derived or publicly derived. There are three visibility modes. They are,

- a. Private
- b. Public
- c. Protected

19. What are the types in inheritance?

The types in inheritance are,

- a. Single inheritance
- b. Multiple inheritance
- c. Multilevel inheritance
- d. Hierarchical inheritance
- e. Hybrid inheritance

20. What is a scope resolution operator?

Scope resolution operator is used to uncover the hidden variables. It also allows access to global version of variables. Scope resolution operator is used to define the function outside the class.

Syntax:

return type <class name> :: <function name>

Example:
#include<iostream. h>

```
int m=10; // global variable m
void main ( )
{
```

```
    int m=20; // local variable m
    cout<<"m="<<m<<"\n";
    cout<<":: m="<<:: m<<"\n";
```

```
}
```

Output:

20

10 (:: m access global m)

Unit III ADVANCED PROGRAMMING

1. What are the various traditional error handling methods?

The various traditional error handling methods are,

- i. Returning error number.
- ii. Global flag manipulation.
- iii. Abnormal termination.

2. What is the importance of exceptional handling?

The importance of exceptional handling is,

- i. Divide the error handling.
- ii. To provide unconditional termination and programmer preferred termination
- iii. For separating error reporting and error handling.
- iv. To solve the object destroy problem.

3. What are the three keywords used for exception handling mechanism?

The three keywords used for exception handling mechanism are,

- i. try → for indicating program area where the exception can be thrown.
- ii. throw → for throwing an exception.
- iii. catch → for taking an action for specific exception.

4. What is the use of unexpected function?

The unexpected function is called when a function throws an exception not listed in its exception specification. Internally, it calls the terminate function to terminate the program execution. The function set_unexpected () is used to call our own unexpected function in place of the built-in unexpected function.

5. What are the challenges in the exception handling mechanism?

The challenges in the exception handling mechanism are,

1. Finding proper catch block.
2. Finding proper catch block for polymorphic object.
3. Backtracking till the beginning of the try block.

6. Why Templates are used in C++?

The Templates are used to develop reusable software component such as functions, classes, etc. Templates allow the construction of a family of templates functions and classes to perform the same operations on different data types.

7. What are rules for invalid function template declaration?

The rules for invalid function template declaration are,

- i. No-argument template function.
- ii. Template–type argument unused.
- iii. Usage of partial number of template arguments.

8. Write the syntax for function Template.

The syntax for function Template is,

```
Template <class T, .....>
Return Type Fun_Name (arguments)
{
    ..... // body of the template
}
```

9. What are the error handling function in C++?

The error handling function in C++ is,

- i. eof()
- ii. fail()
- iii. bad()
- iv. good()

10. What are the rules for virtual function?

The rules for virtual function are,

- i. They cannot be static members.
- ii. They are access by using object pointers.
- iii. A virtual function can be a friend of another class.

11. What are Streams?

Stream is a mechanism, which supplies the input data to a program and presents the processed data in the desired form.

12. What are the file stream classes in C++?

The file stream classes in C++ are,

1. filebuf
2. fstreambase
3. ifstream
4. ofstream
5. fstream
- 6.

13. What is the Stream objects used in C++?

The Stream objects used in C++ is,

- i. cin
- ii. cout
- iii. cerr
- iv. clog

14. List out the ways of Class Template inheritance.

The ways of Class Template inheritance are,

- i. Derive a class template from a base class, which is a template class
- ii. Derive a class template from the base class, which is a template class, add more template members in the derived class
- iii. Derive a class template from a base class which is not a template class, and add template members to that class
- iv. Derive a class template from a base class which is a template class and restrict the template feature.

15. What are the three formatted I/O operation classes?

The three input/output operation classes are,

- i. istream – does formatted input.
- ii. ostream – does formatted output.
- iii. iostream - does formatted input and output.

16. What are the functions of ios classes?

The functions of ios classes are,

Function	Task Performed
width()	Specifies the required number of fields to be used while displaying the output value.
precision()	Specifies the number of digits to be displayed after the decimal point
fill()	Specifies a character to be used to fill the unused area of a field.
setf()	Sets format flag that control the form of output display
unset()	Clears the specified flag

17. What are manipulators?

Manipulators are special functions that are specifically designed to modify the working of a stream.

They can be embedded in the I/O statements to modify the form parameters of a stream. All the predefined manipulators are defined in the header file `iomanip.h`.

18. What are the modes used file stream class?

The modes used file stream class is,

- `in` – open for reading
- `out` – open for writing
- `app` – all write occur at end of file
- `trunc` – truncate the file if already exists
- `nocreate` – open fails if file does not exists
- `noreplace` – open fails if file already exists
- `binary` – open as a binary file

19. What is `get ()` and `put ()` functions?

`get()` is a member function of a stream class `fstream` and is used to read a single character from a file.

`put()` is a member function of a stream class `fstream` and is used to write a single character to the file.

20. What are two types of file pointers used in C++?

The two types of file pointers used in C++ are,

- i. `get` pointer – specifies a location from where the current reading operation is initiated
- ii. `put` pointer - specifies a location from where the current writing operation is initiated

Unit IV OVERVIEW OF JAVA

1. What is Java?

Java is a high-level, third generation programming language, like C, FORTRAN, Smalltalk, Perl, and many others. You can use Java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do.

2. What are the features of Java?

The features of Java are,

- i. Simple.
- ii. Object Oriented.
- iii. Platform Independent.
- iv. Robust.
- v. Multithreaded.
- vi. Secure.

3. What are the various applications of Java?

The various applications of Java are,

- i. Applets
- ii. Networking
- iii. Internationalization
- iv. Security
- v. Object serialization
- vi. Java Database Connectivity (JDBC)

4. What is meant by virtual machine?

A Java virtual machine (JVM), an implementation of the Java Virtual Machine Specification, interprets compiled Java binary code (called bytecode) for a computer's processor (or "hardware platform") so that it can perform a Java program's instructions

5. What are the two components of Java platform?

The two components of Java platform are,

- i. The Java Virtual Machine
- ii. The Java Application Programming Interface (*API*)

6. What is bytecode in Java?

Java bytecode is the form of instructions that the Java virtual machine executes. Each bytecode opcode is one byte in length, although some require parameters, resulting in some multi-byte instructions. Not all of the possible 256 opcodes are used.

7. What is an Object?

An object consists of data and functions known as *methods* which use or change the data. (Methods are similar to procedures or functions in other languages.) Objects of the same kind are said to have the same type or be in the same class. A *class* defines what data can be in an object, and what operations are performed by the methods. One or more objects can be created or "instantiated" from a class.

8. What is an Object and how do you allocate memory to it?

Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

9. What are different types of access modifiers?

The different types of access modifiers are,

- i. public: Any thing declared as public can be accessed from anywhere.
- ii. private: Any thing declared as private can't be seen outside of its class.
- iii. protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages.
- iv. default modifier: Can be accessed only to classes in the same package.

10. What is method overloading and method overriding?

Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding: When a method in a class having the same method name with same arguments is said to be method overriding.

11. List out the primitive types in Java.

The seven primitive types are listed in the following table:

Type	Definition
byte	one-byte signed two's complement integer
short	two-byte signed two's complement integer
int	4-byte signed two's complement integer
long	8-byte signed two's complement integer
float	4-byte IEEE 754 single-precision float
double	8-byte IEEE 754 double-precision float
char	2-byte unsigned Unicode character

12. What is String?

A String is a class used to store a sequence of characters in Java. Strings are constant. Their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared.

Example: String str = "abc";

13. What is an array?

An array is a special object containing a group of contiguous memory locations that have the same name and the same type and a separate variable containing an integer constant equal to the number of array elements. The elements of Java arrays are numbered starting from 0.

Example: double x []; // create an array reference
 x = new double [5]; // create array object

14. What are the methods used in String class?

The methods used in String class are,

- i. charAt(int index)
- ii. compareTo(String anotherString)
- iii. concat(String str)
- iv. copyValueOf(char[] data)
- v. equals(Object anObject)

15. . Why Java use Unicode?

Java use Unicode to represent a character. Unicode defines a fully international character set that can represent all of the characters found in all human languages

16. What is Classpath?

The Classpath is an argument we can set either on the command-line, or through an environment variable that tells the Java Virtual Machine where to look for user defined classes and packages when running Java programs.

17. . What is Garbage collection?

The Garbage collection is the process that is used to free the memory of the objects that are no longer in use. When a program stops referencing an object, it is not required any more and can be deleted. The space that is used by the object is released for use by another object

18. . What are Nested classes?

A nested class is a class defined as a member of another class. The scope of nested class is bounded by the scope of its enclosing class. The nested class has access to the members of its enclosing class including private members.

19. . What do you mean by inheritance?

A subclass inherits variables and methods from its superclass and all of its ancestors. The subclass can use these members as is, or it can hide the member variables or override the methods.

20. . What are the advantages of inheritance?

The advantages of inheritance are,

- i. It permits code reusability.
- ii. Reusability saves time in program development.
- iii. It encourages the reuse of proven and debugged high-quality software, thus reducing problem after a system becomes functional.

21..What Members Does a Subclass Inherit?

A subclass inherits all of the members in its superclass that are accessible to that subclass unless the subclass explicitly hides a member variable or overrides a method. Note that constructors are not members and are not inherited by subclasses.

STICE

Unit V- EXCEPTION HANDLING

1.What are packages?

A Java package is a mechanism for organizing Java classes into namespaces similar to the modules of Modula. Java packages can be stored in compressed files called JAR files, allowing classes to download faster as a group rather than one at a time.

2. List out the Java packages.

The Java packages are,

- i. java.lang
- ii. java.io
- iii. java.awt
- iv. java.net
- v. java.applet
- vi. java.rmi

3.What are Encapsulation, Inheritance and Polymorphism?

- i. Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse.
- ii. Inheritance is the process by which one object acquires the properties of another object.
- iii. Polymorphism is the feature that allows one interface to be used for general class actions.

4.What is the use of super keyword?

A Java(TM) programming language keyword used to access members of a class inherited by the class in which it appears.

5.What do mean by overriding methods?

The ability of a subclass to override a method in its superclass allows a class to inherit from a superclass whose behavior is "close enough" and then supplement or modify the behavior of that superclass.

6.List out the various types of inheritance.

The various types of inheritance are,

- i. Simple inheritance - One base class and one derived class
- ii. Multilevel inheritance - a sub class derived from another sub class
- iii. Hierarchical inheritance - two or more sub class have the same base class

7.Define Interface.

An Interface is defined as a named collection of method definitions (without implementations). An interface can also declare constants. All the methods declared in the interface are abstract methods by default. And all the data members are static final members.

8. How multiple inheritance is implemented in java?

A class can implement more than one interface (the Java platform supports multiple inheritance for interfaces), so the `implements` keyword is followed by a comma-separated list of the interfaces implemented by the class.

9. What is an Exception?

An Exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions during the execution of a program.

10. . How an Exception is handled in java?

A program can catch exceptions by using a combination of the `try`, `catch`, and `finally` statements. The `try` statement identifies a block of code in which an exception can occur. The `catch` statement identifies a block of code, known as an exception handler that can handle a particular type of exception.

11. . What is the use of finally block?

The `finally` statement identifies a block of code that cleans up regardless of whether an exception occurred within the `try` block. A `try` statement must be accompanied by at least one `catch` statement or a `finally` statement and may have multiple `catch` statements.

12. . How Threads are created in Java?

Threads are created in two ways,

- i. extending the `Thread` class
- ii. implementing `Runnable` interface.

13. What is Thread pool?

A Thread pool is a managed collection of threads that are available to perform tasks. Thread pools usually provide,

- i. Improved performance when executing large numbers of tasks due to reduced per-task invocation overhead.
- ii. A means of bounding the resources, including threads, consumed when executing a collection of tasks.

14. . Define Dead Lock.

A Dead Lock is defined as a special type of error that relates specifically to multitasking is dead lock, which occurs when two threads have a circular dependency on a pair of synchronized objects.

15. What do you mean by Thread Scheduling?

Execution of multiple threads on a single CPU in some order is called Thread scheduling. The Java runtime environment supports a very simple, deterministic scheduling algorithm called *fixed-priority scheduling*. This algorithm schedules threads on the basis of their priority relative to other `Runnable` threads.

16. What is life cycle of thread?

New born state
Runnable state
Running state
Blocked state
Dead state

17. What is the use of Data input and output streams?

Data input and output Streams are used to read or write primitive data types in a machine-independent format.

18. What is the use of Buffered Streams?

Buffered streams, buffer data while reading or writing, thereby reducing the number of accesses required on the original data source. Buffered streams are typically more efficient than similar nonbuffered streams and are often used with other streams.

19. What is filtered Stream?

A filtered stream filters data as its being read from or written to the stream. The java.io package provides a set of abstract classes that define and partially implement filter streams.

20. Name two super classes used in byte stream.

The InputStream and OutputStream classes.

CS6456 Object Oriented Programming Question Bank - 16 Marks Questions**PART B****Unit I**

1. Illustrate the function overloading concept with a C++ program to find the volume of cube and cylinder
2. Explain the basic concepts of oops with suitable Examples
3. Write a C++ program to construct student mark list for three subjects. Write the program to display name, rollno, marks, avg and total. Use class and objects.
4. Explain the control structures in C++
5. Define Call by reference and Return by reference.

Unit II

1. Explain the Friend function concept with an example program.
2. What is Dynamic Initialisation of objects? Give a program to illustrate your answer.
3. Explain the constructor concept with its types with example programs.
4. Explain the concept of objects as functional arguments.
5. Write a C++ program to manage a bank account using classes and objects.

Unit III

1. Explain virtual function concept with a program to find the distance between two objects.
2. Explain the Inheritance types with example programs for last three types.
3. What do you mean by operator overloading? What are its types? Explain the types with eg programs.
4. Explain single and multiple Inheritance with an example of your own.
5. Describe manipulation of strings with overloading the following operators: <<, >>, +, -

Unit IV

1. Explain Method overriding in Java with an eg.
2. Construct the pictorial representation of Java Virtual Machine.
3. Write a program in java using constructor concept.
4. What are the different statements and its use in java?
5. Illustrate Inheritance in Java with suitable program.
6. Give an explanatory answer to define the difference between Java and C++, Characteristics of Java and the concepts in java.

Unit V

1. Explain the Life cycle of Thread with an eg
2. Describe package concept to perform arithmetic operations. Explain how to use it?
3. Explain the different states in Life cycle of applet?
4. Define Interfaces? Explain the extension of interfaces, implementation and accessing it.
5. What are the Different Exceptions caught, Explain the types with eg program.
6. Explain try, catch and finally statements with eg.