MODULE 1

1. What are the different programming paradigms?
   Structured Programming
   Procedural Programming
   Object Oriented Programming

2. What is Procedural Programming?
   It is a programming paradigm derived from structured programming
   It is based on the concept of procedure call.
   Procedures, also known as routines, subroutines, or functions, simply contain a series of computational steps to be carried out.
   Eg. C, C++, pascal, fortran

3. What is Object Oriented Programming?
   Object-oriented programming (OOP) is a programming language model in which programs are organized around data, or objects, rather than functions and logic.
   • An object can be defined as a data field that has unique attributes and behavior.
   • Example:Java, Python, JavaScript

4. Difference between C, C++, Java
   Module 1, Slide 10.
   Points:
   Programming Paradigm
   Pointers
   Preprocessor directives
   Memory Allocation
   DB connectivity
   Code Translation(Compiled/Interpreted)
   Complex Data Types
   String Type
   Operator Overloading
   Multithreading
   Inheritance

5. What is Java?
   Java is a programming language and a platform.
   Java is a high level, robust, object-oriented and secure programming language.

6. Features of Java
   Simple
   Object Oriented
   Portable
   • Platform independent

- Secured
- Robust
- Architecture neutral
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic

Details of each are in Mod 1 PPT

7. What is an Object?
   An object is an instance of a class. It is both a physical as well as logical entity. They have three characteristics:
   State: represents data of an object (class variables)
   Behavior: Represents functionality of the object (methods)
   Identity: Object identity is typically a unique id which is not shown to the user but used internally by the JVM.

8. What is a Class?
   A class is a group of objects that have common properties. It is a template from which objects are created. It is a logical entity. Classes can contain fields, methods, constructors, blocks, nested class and interface.

9. The 4 basic properties of OOP
   Inheritance:
   Inheritance is a mechanism in which a child object/class acquires all the properties of the parent object/class.
   It is useful because it helps code reusability and thus prevents the need of rewriting code. You can create new classes based on existing classes so we can reuse the methods and fields of the parent class while defining new methods and fields in the child class.
   Inheritance represents IS-A relationship or parent-child relationship

   Polymorphism:
   Polymorphism is a concept by which a single action can be performed in multiple ways. There are two types of polymorphism in java, compile time polymorphism and runtime polymorphism.

   Compile time polymorphism is achieved by method overloading. In method overloading, multiple methods with the same name but different parameter lists are defined within the same class.
   The parameter lists may differ by number of arguments or types of arguments or both.
   Eg.

void hello(int a)
void hello(float a)
void hello(int a, int b)

Runtime polymorphism is achieved through method overriding. In method overriding, a subclass provides a specific implementation for a method that is already defined in its superclass.

Abstraction:
Abstraction is a way of hiding the implementation details from the user and providing only a well defined interface, i.e. showing only the functionality and hiding the internal details.

Encapsulation:
Encapsulation in Java is a process of wrapping data and function together in a single unit. In java this is achieved through classes.

10. How is java platform independent?

    Java is often referred to as platform-independent because of its "Write Once, Run Anywhere" (WORA) capability.
    Java Code is compiled into an intermediate form called bytecode. This bytecode can be executed on any platform(Windows, Linux, Mac etc.) on the Java Virtual MAchine(JVM).

11. What is multithreading in java?

    Multithreading in Java refers to the capability of a program to execute multiple threads concurrently within a single process. A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads.
    • The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area.

12. Why is java robust?
    It has strong memory management.
    It has no pointers which helps avoid security problems
    There is an exception handling mechanism in java.

Module 2:

1. What are classes and objects?
   Refer above questions.

2. What is a constructor?
   A constructor initializes an object when it is created.
   It has the same name as the class name and has no return type.
   A constructor is used to give initial values to the instance variables defined by the class and perform any other startup procedures required.
   All classes have constructors, whether you define one or not as java automatically provides a default constructor that initializes all the variables to 0.

3. What are the types of constructors in java?
   There are two types of constructors:
   1. Default Constructor:
      Default constructor provides the default values to the object like 0, NULL etc. If no constructor is defined for a class, default constructor is used to initialize the object.
   2. Parameterised Constructor:
      It is a constructor that has parameters. It is used to provide different values to different objects.

4. What is constructor overloading?
   Constructor overloading in java is a technique of having more than one constructor with different parameter lists, where each performs separate tasks. The compiler differentiates them on the basis of the number of parameters in the list as well as their types.

5. What is a method?
   A method in java is a collection of statements grouped together to perform a specific operation.
   Syntax:
   modifier returnType nameOfMethod (Parameter List)
   { // method body }

6. Differentiate between constructor and method

Si no	Constructor	Method
1	Constructor is used to initialize the **state** of an object.	Method is used to expose **behavior** of an object.
2	Constructor must not have return type	Method must have return type.
3	Constructor is invoked implicitly.	Method is invoked explicitly.
4	The java compiler provides a default constructor if you don't have any constructor.	Method is not provided by compiler in any case.
5	Constructor name must be same as the class name	Method name may or may not be same as class name.

7. What is a destructor?
   A destructor is a special member function of a class that is executed whenever object goes out of scope or whenever the delete operation is applied to the pointer of an object.
   It has the same name as the class name and has the tilde(~) as a prefix.
   It has no arguments or return types.

   In java the finalise method is equivalent to the c++ concept of destructor. The process of removing an object from the running program is known as garbage collection.

8. What are access specifiers/modifiers in java?
   The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

    1. **Public**: Class, Method, Constructor or Interface which are public can be accessed from any other class in the program, within the package and even outside the package.
    2. **Protected**: The access level of methods, variables and constructors declared protected is all classes within the package and child classes outside the package. It cannot be accessed from outside the package without making a child class. Cannot be applied to class and interface.
    3. **Private**: Methods, Variables and Constructors which are private can only be accessed from within the class, classes and interfaces cannot be private.
    4. **Default**: Default access modifier is applied when we do not explicitly declare an access modifier. The variable or method which has default access level can be accessed from any other class within the same package.

Access Modifiers	Default	private	protected	public
Accessible inside the class	yes	yes	yes	yes
Accessible within the subclass inside the same package	yes	no	yes	yes
Accessible outside the package	no	no	no	yes
Accessible within the subclass outside the package	no	no	yes	yes

9. What are the ways to read input in Java?
   Command Line Interpreter
   Buffered reader Class
   Scanner Class

10. What is a static variable in Java?
When a variable is declared with keyword static it is called a class variable.
All instances of the class share the same variable, a class variable can be accessed without the need of creating an object which makes it memory efficient.

11. What is a static method in java?

A static method belongs to the class rather than an object of the class.
Static method can be called without the need of creating an object of the class.
Static method can access the static data member and can change the value of it.

Disadvantages:
The static method can not use non static data member or call a non static method.
You can not use keywords “this” and “super”

12. Operators in java

There are many types of operators in java which are given below:

- o Unary Operator +, -, ++, --
- o Arithmetic Operator +,-,/,*
- o Shift Operator <<, >>
- o Relational Operator >, >=, <, <=, ==, !=
- o Bitwise Operator &, !, |, ^
- o Logical Operator &&, ||, !
- o Ternary Operator ?:
- o Assignment Operator =, +=, -=

13. What are arrays in Java?

• Array is a collection of similar type of elements that have contiguous memory location. Java array is an object the contains elements of similar data type. We can store only fixed set of elements in a java array.

- • In java, array is an object.
- • There are two types of array.
- • Single Dimensional Array
- • Multidimensional Array

Module 3:

1. What are arrays in Java?
   • Array is a collection of similar type of elements that have contiguous memory location. Java array is an object the contains elements of similar data type. We can store only fixed set of elements in a java array.
   • In java, array is an object.
   • There are two types of array.
   • Single Dimensional Array
   • Multidimensional Array

2. What is Collection Framework?
   Collection framework provides an architecture to store and manipulate the group of objects. It helps to achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

3. What are vectors in c++?
   Vectors are a dynamic array of objects which can store heterogenous objects.
   They are useful when you dont know the exact size of array in advance as they can grow and shrink as necessary.
   We cannot store primitive data types in array they must be converted to objects, as vector can only store objects.

ArrayList	Vector
1) ArrayList is **not synchronized.**	Vector is **synchronized.**
2) ArrayList **increments 50%** of current array size if the number of elements exceeds from its capacity.	Vector **increments 100%** means doubles the array size if the total number of elements exceeds than its capacity.
3) ArrayList is **not a legacy** class. It is introduced in JDK 1.2.	Vector is a **legacy** class.
4) ArrayList is **fast** because it is non-synchronized.	Vector is **slow** because it is synchronized, i.e., in a multithreading environment, it holds the other threads in runnable or non-runnable state until current thread releases the lock of the object.
5) ArrayList uses the **Iterator** interface to traverse the elements.	A Vector can use the **Iterator** interface or **Enumeration** interface to traverse the elements.

1. What are wrapper classes?

   A Wrapper class in Java is a class whose object wraps or contains primitive data types. They are used to convert primitive data types to objects. It is useful because classes in the collection framework only allow objects as data type and not primitive data types, so Wrapper classes help us overcome this problem.

2. What is autoboxing and unboxing?

   The process of automatically converting a primitive data type into the object of its corresponding wrapper class is called autoboxing.

   The process of automatically converting a Wrapper Class Object into its corresponding primitive data type is called unboxing.