31/07/25 - Assignment

1. class

-> class is an template / blueprint for creating an objects

-> class is declared using class keyword

-> Eg: class Main { }

-> Inside class we can declare variables, methods and constructors

2. global and local variables

-> variables which declared inside any methods is called local variables, it can be accessible only inside that methods

-> variables which declared outside any methods is called global variables, it can be accessed anywhere

-> It is also called as fields

-> There are two types of global variables

1) static

2) nonstatic (Instance)

3. Data types

There are two types of datatypes in java. They are

1. primitive data types (fixed)

2. Non-primitive datatypes (mutable)

primitive data types

-> byte, Short, int, long, float, double, char, boolean

-> In default we use int and double only

Non-primitive datatypes

-> All classes, String, arrays, ...

4. Packages

-> packages are like folders used for organizing classes. There are two types of packages.

1. Built-in packages (packages from java API)-C

2. User-defined packages (own packages)

→ we can create package using-(package packname)

→ compile using javac -d . classname

→ run using java package name. classname

5. object

→ Object is an Instance of an class, it has an both state and behaviour

→ we can create objects using new Keyword

→ Syntax - className objectname = new classname();

6. Constructor and constructor chaining

→ Constructor have an same name as class name and it does not have any return type it is used for intializing object specific values

→ Evdy time we create an object an constructor is called, and it is default constructor

→ Jvm has one default constructor

→ Constructor chaining is where one constructor calls another constructor with the help of this() keyword, this() must be an first Statement

7. Inheritance

-> An object of one class acts an object of another class is called inheritance

-> we can use extends keyword in child class

-> parent class is also called super class(or) base class

-> child class is also called subclass(or) derived class

-> for code reusability we can use Inheritance

8. Abstraction

-> Showing only necessary data and hiding unwanted Information to the user

-> use abstract keyword in method and in className

-> we cannot create object for abstract class, it will be created only in child class

-> abstract keyword is non-access modifier

-> abstract methods cannot have body and it is provided by subclass

9. polymorphism

→ polymorphism means one Interface many forms

. There are two types of Polymorphism. They are

1. compile time polymorphism (method overloading)

2. run time polymorphism (method overriding)

→ methods having same name with different number of arguments and different types of arguments in same class is called method overloading

→ methods having same name with same number of arguments in different classes with parent and child relation is called method overriding

b. Encapsulation

→ It is an process of binding data and code together, to achieve this we can declare variables and attributes as private and use public setter and getter method to access and update the value of that variable (c)

→ It increase the security of the data

11. Interface

-> Another way of achieving abstraction

-> Interface is not a class it is an rules

-> Interface methods are by-default abstract

-> Implements keyword is used Instead of Extends keyword, it does not have object

-> only main differente between 100% abstract class and Interface is it doesnot have Constructor

12. this vs super

-> this is the keyword and it refers to the current object in a method

-> Super is the keyword and it refers to the parent class object's and methods

13. this () vs super()

-> this() is used for calling a constructor from another constructor

-> this() must be an first statement in constructor

-> super() is used for calling the parent class constructor like this() , super() also be in first statement

14. final keyword

-> final keyword can be used in class, methods, variables

-> If we use final keyword in class, they cannot be inherited by other classes

-> If we use in methods, it does not support overriding

-> If we use in variables, we cannot change that value of the variables

15. static vs non static

-> static refers to class specific Information class Information is common for all

-> Static keyword is used for declaring static variables

-> Static variables can be accessed using classname and objectname

→ non-static refers to object Specific Information

→ non-static variable is not used Inside static method, but we can use with help of object creation

→ memory is allocated everytime when an new object is created