DAY – 10

# JAVA:

****POLYMORPHISM:****

It is a concept where we can perform a single action in different ways.

Poly => many

Morphishm => morphs => forms

Advantages:

* Code reusability
* Flexibility & Extensibility
* Dynamic method Invocation
* Interface Implementation
* Method overloading
* Reduce code complexity

Types:

1. Compile-time Polymorphism (Static binding)

Achieved through method overloading

1. Run-time Polymorphism (Dynamic binding)

Achieved through method overriding

****SUPER KEYWORD IN JAVA:****

The Super keyword in java is a reference variable i.e.; used to immediate parent class objects.

Advantages:

* Simplifies constructor chaining
* Access to overridden method
* Access to hidden files
* Code reusability

****FINAL KEYWORD IN JAVA:****

* The final keyword in java is used to restrict the user.
* It is also known as a non-access modifier.
* final class can’t be inherited.
* The final method can be inherited, but we can’t override it.
* We can use final keyword with:
* Variable
* Method
* Class
* Parameter

****STATIC & DYNAMIC BINDING:****

****STATIC:****

* When the type of object is determined at compile time.
* If there is any private, final, or static method in a class, there is a static binding.

****DYNAMIC****

* When the type of object is determined at run time.

****PACKAGES IN JAVA:****

A Java package is a group of similar types of classes, interfaces, and sub-packages.

Example:

Package myPackage;

Public class Main

{

Public static void main (String [] args)

{

System.out.println (“Hi”);

}

}

****ENCAPSULATION IN JAVA:****

* It is a process of wrapping code and data together into a single unit.
* Encapsulation helps to protect data & control access to it.
* It protects sensitive data from being accessed directly.
* It hides unnecessary data from the user of a class and only shows the functionality of the class.
* Changes can be made internally without affecting the external interface.
* Encapsulated classes can be reused for projects.

Example:

Public class Student

{

public string name;

public string getName()

{

return name;

}

public void setName()

{

This.name=name;

}

}