PW ASSIGNMENT (OOPS Assignment)

QUESTION → What is inheritance in java?

Answer → Taking the functionality of A class by using the extends keyword , a child class can inherite everything from its parent class , is known as Inheritance in java.

QUESTION → What is super class and subclass in java ?

Answer → a class is known as subclass , when it inherits properties from its parent class by using extends keyword.

A class whose properties and behaviour (data member & member function) is inherited by other class by using extends keyword is known as Super class.

QUESTION → How Inheritance is implemented/achieved in java ?

Answer → By using extends keyword Inheritance is achieved.

QUESTION → What is Polymorphism in java?

ANSWER → When two methods having the same name , but having different parameters are called method overloading. Example-->

class Multiply{

```
public int multiply(int a, int b)
    {
        int prod = a * b;
        return prod;
    }
public int multiply(int a, int b, int c)
    {
        int prod = a * b * c;
        return prod;
    }
}
```

```
Class Main{
public static void main(String[] args){
Multiply mul = new multiply();
Mul.multiply(10,5);            //this will give op as -->50
Mul.multiply(10,10,10);        // this will give op as -->1000
}
}
```

QUESTION → Differentiate between method overloading and method overriding in java?

Answer → here is the difference.

Overriding	Overloading
Implements "runtime polymorphism	Implements "compile time polymorphism"
The method call is determined at runtime based on the object type	The method call is determined at compile time
Occurs between superclass and subclass	Occurs between the methods in the same class
Have the same signature (name and method arguments)	Have the same name, but the parameters are different
On error, the effect will be visible at runtime	On error, it can be caught at compile time

QUESTION → What is an abstraction in java?

Answer --> Hiding the internal working mechanism is called abstraction in java .

```
Abstract class PWSKILL{
abstract void m1();
}
```

QUESTION → What is the difference between abstract method and final method in java?

Answer --> final method --> in java final could be method , class or a variable .

Final method , variable cannot be redeclared. Whereas final class cannot be inherited.

Abstract method -->> the method which is defined as abstract cannot have the body of that method, body of abstract method is defined in its implementation class.

QUESTION → What is the difference between abstraction and encapsulation in java?

Answer -->

Abstraction	Encapsulation
Abstraction is a feature of OOPs that hides the unnecessary detail but shows the essential information.	Encapsulation is also a feature of OOPs. It hides the code and data into a single entity or unit so that the data can be protected from the outside world.
It solves an issue at the design level.	Encapsulation solves an issue at implementation level.
It focuses on the external lookout.	It focuses on internal working.
It can be implemented using abstract classes and interfaces.	It can be implemented by using the access modifiers (private, public, protected).
It is the process of gaining information.	It is the process of containing the information.
In abstraction, we use abstract classes and interfaces to hide the code complexities.	We use the getters and setters methods to hide the data.
The objects are encapsulated that helps to perform abstraction.	The object need not to abstract that result in encapsulation.

QUESTION → What is the difference between runtime polymorphism & compile time polymorphism in java?

Answer -->

Compile Time Polymorphism	Runtime Polymorphism
Compile time polymorphism is less flexible as all things execute at compile time.	Run time polymorphism is more flexible as all things execute at run time.
In Compile time Polymorphism, the call is resolved by the compiler.	In Run time Polymorphism, the call is not resolved by the compiler.
Inheritance is not involved.	Inheritance is involved.
It is also known as Static binding, Early binding and overloading as well.	It is also known as Dynamic binding, Late binding and overriding as well.
It provides fast execution because the method that needs to be executed is known early at the compile time.	It provides slow execution as compared to early binding because the method that needs to be executed is known at the runtime.
Method overloading is the compile-time polymorphism where more than one method shares the same name with different parameters or signature and different return type.	Method overriding is the runtime polymorphism having the same method with same parameters or signature but associated with compared, different classes.