TechWeeks

Day5: Topics for today!!!!

Inheritance and **Polymorphism**

So far you all have got a hang of Java i assume , now for today you guys will learn about **Inheritance** , **Encapsulation** , **Polymorphism** etc

- INHERITANCE : it is the pillar of OOP
  - It can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.
  - So , in simple terms the **child** class has its **own** methods and properties but it also acquires the properties and methods of **Parent** class
  - **Syntax**: here extends is the keyword to extend the Sub(child) that inherits properties of super (parent class)

```
class Super {
   .....
}
class Sub extends Super {
   .....
 }
```

- I highly recommend you all to research about inheritance below I have provided few video link you can watch to understand inheritance .
- https://www.youtube.com/watch?v=_9R_in_7cT4
- https://www.youtube.com/watch?v=7dwBc-ZZEYg

TechWeeks

Exercise for Inheritance :

- The Exercise codes should be solved and the program should be uploaded on our **Github** for us to check your Exercise.

1. Create a Super Class Employee and a Sub Class User which extends Employee
   - Employee class has variable salary float type (4000)
   - User class has variable bonus type(1000)
   - Print both salary and bonus of User and Employee

2. Write a Code with the guidelines provided below
   - Create a Parent Class : Animal
   - Dog is a Subclass of Animal
   - Cat is Subclass of Dog
   - Create a Separate class and call all the above class by creating a Instance of Cat class inside it

3. Find the Reason behind Java not supporting Multiple Inheritance!

4. Copy the code given below and run in Netbeans IDE

```
class Superclass {
    int age;
    Superclass(int age) {
        this.age = age;
    }
    public void getAge() {
        System.out.println("The value of the variable named age in super
class is: " +age);
    }
}
public class Subclass extends Superclass {
    Subclass(int age) {
        super(age);
    }
    public static void main(String args[]) {
        Subclass s = new Subclass(24);
        s.getAge(); }}
```

❖ **Polymorphism** in Java:

1. Polymorphism is that in which we can perform a task in **multiple** forms or ways. It is applied to the functions or methods. **Polymorphism** allows the object to decide which form of the function to implement at **compile-time** as well as **run-time**.
2. So , basically there are 2 types of Polymorphism :
   - Compile-time polymorphism (**Method overloading**)
   - Run-time polymorphism (**Method Overriding**)

3. Method Overloading : If a class has multiple methods having same name but different in parameters, it is known as **Method Overloading**.
   - Method Overloading helps to increase readability of your program.
   - Two ways to overload a Method ie:
     - By changing number of arguments.
     - By changing the data type.

4. Example of Method Overloading :-
   - In this code add method add has 2 parameter and again a add method had 3 parameter this helps user to keep name of methods same but control the output of method by providing input as per parameters

```
class Adder{
static int add(int a,int b){return a+b;}
static int add(int a,int b,int c){return a+b+c;}
}
class TestOverloading1{
public static void main(String[] args){
System.out.println(Adder.add(11,11));
System.out.println(Adder.add(11,11,11));
}}
```

5. Method Overriding : If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in Java**.

- Usage of Java Method Overriding
- Method overriding is used to provide the specific implementation of a method which is already provided by its superclass.
- Method overriding is used for runtime polymorphism.

Rules for Java Method Overriding

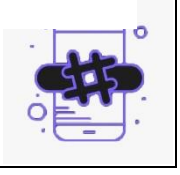
1. The method must have the same name as in the parent class
2. The method must have the same parameter as in the parent class.
3. There must be an IS-A relationship (inheritance).

Example for Method Overriding :

- In this code we see that both class vehicle and child class bike2 has same method run() this is called Method Overriding .

```
  class Vehicle{
 //defining a method
 void run(){System.out.println("Vehicle is running");}
}
//Creating a child class
class Bike2 extends Vehicle{
 //defining the same method as in the parent class
 void run(){System.out.println("Bike is running safely");}

 public static void main(String args[]){
 Bike2 obj = new Bike2();//creating object
 obj.run();//calling method
 }
}
```



TechWeeks

Exercise for Polymorphism:

1. Create a Parent class a1 and a child class a2 which extends a1 using method overloading create a method b1 in both the classes

2. Create a class Addition with 2 methods with same name ie Sum , were one methods takes 2 input and gives summation of those two integer
   The other sum method takes input as 3 integer and gives there summation .