OBJECT ORIENTED PROGRAMMING

Affefah Qureshi

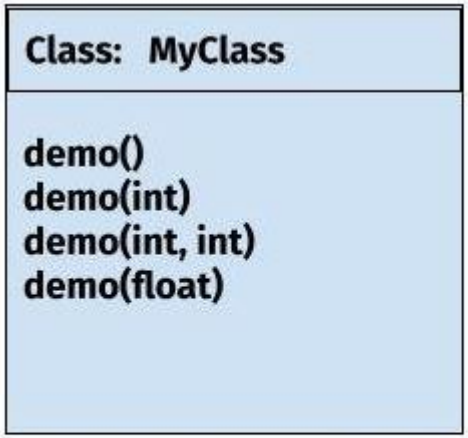
Department of Computer Science

Iqra University, Islamabad Campus.

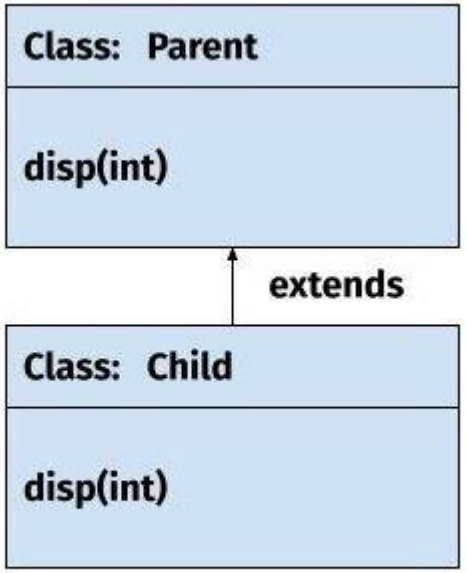
METHOD OVERLOADING VS OVERRIDING

- In Method Overloading , we have multiple methods with same name but with different signatures.
- In **Method Overriding**, we define the same method with the same signature in the child class and change the body of the method.

METHOD OVERLOADING VS OVERRIDING



Method Overloading



Method Overriding

CAN MAIN METHOD IS OVERLOADED?

```
public class JavaExample {
  //Overloading main() methods
  //Here we have three variations of main() method
  public static void main(String[] args)
  {
    System.out.println("Main Method: String[] args");
    //calling main with one string param
    JavaExample.main("AA");
    //calling main with two string param
    JavaExample.main("BB", "CC");
  }
```

CAN MAIN METHOD IS OVERLOADED?

```
// Overloaded main methods
 public static void main(String arg)
 {
   System.out.println("Main Method: String arg");
   System.out.println(arg);
 }
 public static void main(String arg, String arg2)
 {
   System.out.println("Main Method: String arg, String arg2");
   System.out.println(arg+" "+arg2);
 }
}
```

STRING ARGS VS STRING[] ARGS/ STRING ARGS[]

- String... args will declare a method that expects a variable number of String arguments. The number of arguments can be anything at all: including zero.
- String[] args and the equivalent String args[] will declare a method that expects exactly one argument: an array of strings.

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
- Method overriding helps in writing a generic code based on the parent class.
- It provides multiple implementations of the same method and can be used to invoke parent class overridden methods using super keyword. It defines what behavior a class can have.

RULES FOR JAVA METHOD OVERRIDING

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

METHOD OVERRIDING EXAMPLE

```
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
  }
}
```

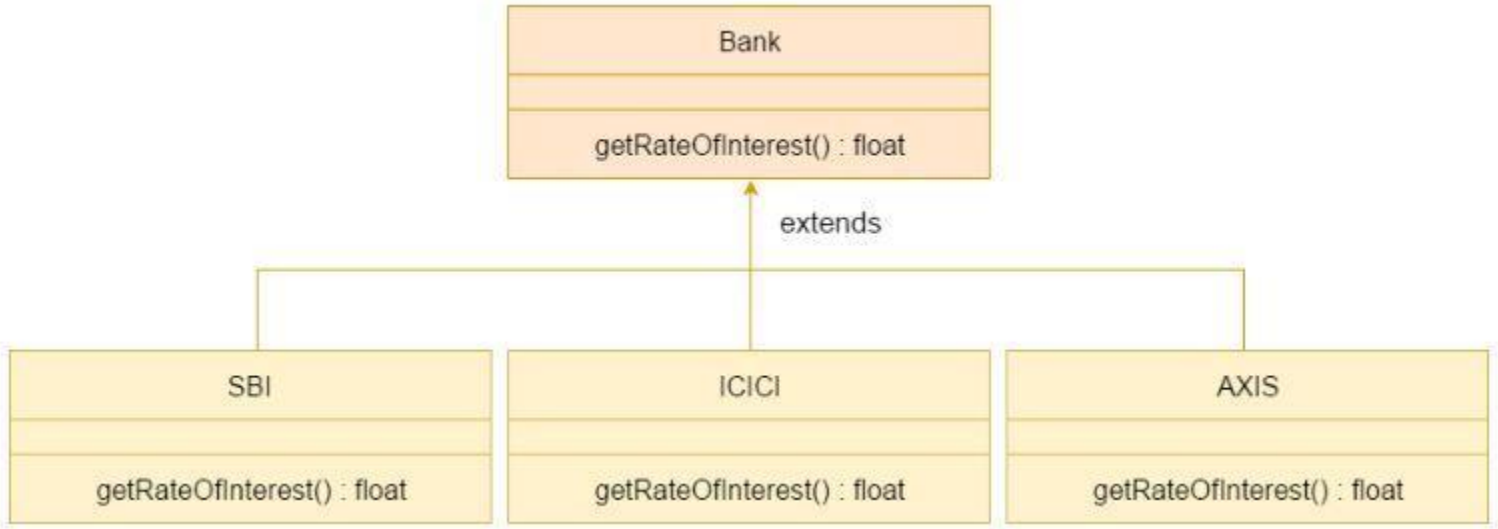
Output:

Bike is running safely.

A REAL EXAMPLE OF JAVA METHOD OVERRIDING

- Consider a scenario where Bank is a class that provides functionality to get the rate of interest.
- However, the rate of interest varies according to banks
  - For example, SBI, ICICI and AXIS banks could provide 8%, 7%, and 9% rate of interest.

METHOD OVERRIDING EXAMPLE



METHOD OVERRIDING EXAMPLE

```
//Java Program to demonstrate the real scenario of Java Method Overriding
//where three classes are overriding the method of a parent class.
//Creating a parent class.
class Bank{
int getRateOfInterest(){return 0;}
}
//Creating child classes.
class SBI extends Bank{
int getRateOfInterest(){return 8;}
}
class ICICI extends Bank{
int getRateOfInterest(){return 7;}
}
```

METHOD OVERRIDING EXAMPLE

```
class AXIS extends Bank{
int getRateOfInterest(){return 9;}  }
//Test class to create objects and call the methods
class Test2{
public static void main(String args[]){
SBI s=new SBI();
ICICI i=new ICICI();
AXIS a=new AXIS();
System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());
System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());
System.out.println("AXIS Rate of Interest: "+a.getRateOfInterest());
} }
```

Output:

SBI Rate of Interest: 8
ICICI Rate of Interest: 7
AXIS Rate of Interest: 9

SUPER KEYWORD IN METHOD OVERRIDING

- The super keyword in Java is used for calling the parent class method or constructor.
- For example, let's consider a method named newMethod() in the parent class, then super.newMethod() can be used to call the newMethod() method of parent class.
- super() can be used to call the constructor of the parent class.
- The superclass and subclass can have attributes with the same name. We use the super keyword to access the attribute of the superclass.

METHOD OVERRIDING EXAMPLE

```
// Class physics
class Physics {
    // method say which is overridden method here
   public void say() {
      System.out.println("This is class Physics");
   }
}
// Class Topic
class Topic extends Physics {
    // method say which is overriding method here
   public void say() {
       // this will call say method of Physics Class
       super.say();
      System.out.println("This is class Topics");
   }  }
```

METHOD OVERRIDING EXAMPLE

```
class PhyMain {
   public static void main(String args[]) {
      Physics a = new Physics(); // Physics reference and object
      Physics b = new Topic(); // Physics reference but Topic
object

      a.say(); // runs the method in Physics class
      b.say(); // runs the method in Topic c
   }
}
```

Output:

This is class Physics
This is class Physics
This is class Topics

METHOD OVERRIDING EXAMPLE

```
class Animal {
  protected String type="animal";
}
class Dog extends Animal {
  public String type="mammal";
  public void printType() {
    System.out.println("I am a " + type);
    System.out.println("I am an " + super.type);   }
}
class AnimalMain {
  public static void main(String[] args) {
    Dog dog1 = new Dog();
    dog1.printType();
  }
}
```

Output:

I am a mammal
I am an animal