Object Oriented Programming

ABSTRACTION

Abstraction

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

- ➢ It shows only essential things to the user and hides the internal details.
  For example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.

Ways to achieve Abstraction

There are two ways to achieve abstraction in java

- Abstract class (0 to 100%)
- Interface (100%)

Abstract Class

- A class which is declared with the abstract keyword is known as an abstract class in Java.
- It can have abstract and non-abstract methods (method with the body).
- It needs to be extended and its method implemented. It cannot be instantiated.

Abstract Class

- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- It cannot be instantiated.
- It can have constructors and static methods also.
- It can have final methods which will force the subclass not to change the body of the method.

Abstract Class

Example of abstract class:

```
abstract class A{}
```

Abstract Method in Java

A method which is declared as abstract and does not have implementation is known as an abstract method.

Example:

```
abstract void printStatus();//no method body and abstract
```

Example of Abstract class that has an abstract method

In this example, Bike is an abstract class that contains only one abstract method run. Its implementation is provided by the Honda class.

```
abstract class Bike{
  abstract void run();
}
class Honda4 extends Bike{
void run(){System.out.println("running safely");}
public static void main(String args[]){
 Bike obj = new Honda4();
 obj.run();
} }
```

Output:

```
running safely
```

Example of Abstract class

```
abstract class Shape{
abstract void draw();
}

class Rectangle extends Shape{
void draw(){System.out.println("drawing rectangle");}
}
class Circle1 extends Shape{
void draw(){System.out.println("drawing circle");}
}

class TestAbstraction1{
public static void main(String args[]){
Shape s=new Circle1();
s.draw();
}  }
```

Output:

```
drawing circle
```

Example of Abstract class

```
abstract class Bank{
abstract int getRateOfInterest();
}
class SBI extends Bank{
int getRateOfInterest(){return 7;}
}
class PNB extends Bank{
int getRateOfInterest(){return 8;}
}

class TestBank{
public static void main(String args[]){
Bank b;
b=new SBI();
System.out.println("Rate of Interest is: "+b.getRateOfInterest()+" %");
b=new PNB();
System.out.println("Rate of Interest is: "+b.getRateOfInterest()+" %");
}}
```

<u>Output:</u>

```
Rate of interest is 7%
Rate of interest is 8%
```

Example of Abstract class

```
class Bike12{
abstract void run();
}
```

Output:

- compile time error

Thank You