JBK1009-Abstraction

1)Simple Abstract class example:

```
public abstract class AbstractClass_Demo {
       public void disp1(){
           System.out.println("Concrete method of abstract class");
        }
        abstract public void disp2();
}
public class Test {
       public void disp2(){
             System.out.println("I'm overriding abstract method");
        }
      public static void main(String[] args) {
             Test obj = new Test();
            obj.disp2();
         obj.disp1();
      }  }
```

2)Create Multiple Abstarct Methods within a Abstract Class in Java

```
public abstract class AbstractClass_Ex2 {
      abstract void get(int a, int b);
      abstract void add();
      abstract void display();
}
class SubClass extends AbstractClass_Ex2 {
      int x, y, z;
```

```
	void get(int a, int b) {
		x = a;
		y = b;
	}
	void add() {
		z = x + y;
	}
	void display() {
	System.out.println("The Addition is : " + z);
	}  }

class MainClass {
	public static void main(String args[]) {
		SubClass obj = new SubClass();
		obj.get(10, 20);
		obj.add();
		obj.display();
	} }
```

3)Create Multiple Abstarct Class & Abstract Methods in Java

```
public abstract class AbstractClass_Ex3 {
	abstract void getName(String name);
	abstract void getGender(String gender);
}
abstract class AbstClass extends
AbstractClass_Ex3 {

	abstract void getCity(String city);
	abstract void getCountry(String country);
}
class SubClass extends AbstClass {
	String name, gender, city, country;
	void getName(String name) {

		this.name = name;
	}
	void getGender(String gender) {

		this.gender = gender;
	}
	void getCity(String city) {

		this.city = city;
	}

	void getCountry(String country) {
		this.country = country;
	}
	void display() {
		System.out.println("Name : " + name);
		System.out.println("Sex : " + sex);
		System.out.println("City : " + city);
	System.out.println("Country : " + country);
	}
}
class MainClass {
	public static void main(String args[]) {
		SubClass obj = new SubClass();
		obj.getName("Balaji");
		obj.getSex("Male");
		obj.getCity("Erode");
		obj.getCountry("India");
		obj.display();
	}  }
```

4) Access Abstarct Class Methods within a Package in Java

```
package balaji.abst;
public abstract class AbstClass {
	abstract void getVal(String name);
	abstract void display();
}
package balaji.abst;
public class MainClass extends AbstClass {
	String name;
	void getVal(String name) {
		this.name = name;
	}
	void display() {
		System.out.println("Hello " + name);
	}
}

class Class {
	public static void main(String args[]) {
```

```
            MainClass obj = new MainClass();

            obj.getVal("SampleCodez");
            obj.display();
      } }
```

5)simple example of interface

```
public interface InterfaceA {
      public void myMethod();
}
public interface InterfaceB {
      public void myMethod();

}
public class InterfaceDemo implements
InterfaceA,InterfaceB{
      public void myMethod()
        {
           System.out.println(" Multiple inheritance
example using interfaces");
        }
      public static void main(String[] args) {
            InterfaceDemo id=new InterfaceDemo();

            id.myMethod();
      }
}
```

6)simple example on inherited interface

```
public interface Interface1 {
       public void method1();

}
public interface Interface2 extends Interface1 {
       public void method2();

}
public class Test implements Interface2 {
      public void method1(){
         System.out.println("Implemented
method1");
       }
       public void method2(){
         System.out.println("Implemented
method2");
       }


      public static void main(String[] args) {
            Test t=new Test();
            t.method1();
            t.method1();
            }
}
```

7)example of interface with variable

```
interface Moveable {
 int AVG-SPEED = 40;
 void move();
}
class Vehicle implements Moveable {
 public void move() {
  System .out. print in ("Average speed is"+AVG-SPEED");
 }
 public static void main (String[] arg) {
  Vehicle vc = new Vehicle();
  vc.move();
 }
}
```

8)interface example:

```
interface Moveable {
 boolean isMoveable();
}
interface Rollable{
 boolean isRollable
}  class Tyre implements Moveable, Rollable {
 int width;

 boolean isMoveable(){
  return true;
 }
 boolean isRollable() {
  return true;
 }
 public static void main(String args[]) {
  Tyre tr=new Tyre();
  System.out.println(tr.isMoveable());
  System.out.println(tr.isRollable());
 } }
```

9) interface example:

```
public interface MyInterface {
      public void method1();
      public void method2();
}
public class XYZ implements MyInterface {

       public void method1(){
          System.out.println("implemented
method1");
        }
        public void method2()  {
          System.out.println("implemented
method2");
        }
        public static void main(String arg[]) {
          MyInterface obj = new XYZ();
          obj. method1();
        }
}
```