

Abstraction

- Simple Abstract class example

```
package com.javabykiran;  
  
public abstract class AbstractclassEx1 {  
  
    public void disp1() {  
        System.out.println("Concrete method of abstract class");  
    }  
  
    abstract public void disp2();  
}
```

```
package com.javabykiran;  
  
public class AbstractTestEx1 extends AbstractclassEx1 {  
  
    public void disp2() {  
        System.out.println("I'm overriding abstract method");  
    }  
  
    public static void main(String[] args) {  
        AbstractTestEx1 obj = new AbstractTestEx1();  
        obj.disp2();  
        obj.disp1();  
    }  
}
```

- Create Multiple Abstract Methods within a Abstract Class in Java

```
package com.javabykiran;  
  
public abstract class AbstractclassEx2 {  
  
    abstract void get(int a, int b);  
  
    abstract void add();  
  
    abstract void display();  
  
}
```

```
package com.javabykiran;  
  
class AbstractSubclassEx2 extends AbstractclassEx2 {  
    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);  
    }  
  
}
```

```
package com.javabykiran;  
  
class AbstractTestEx2 {  
  
    public static void main(String args[]) {  
        AbstractSubclassEx2 obj = new AbstractSubclassEx2();  
        obj.get(10, 20);  
        obj.add();  
        obj.display();  
    }  
  
}
```

- Create Multiple Abstract Class & Abstract Methods in Java

```
package com.javabykiran;  
  
public abstract class AbstractEx3 {  
  
    abstract void getName(String name);  
  
    abstract void getGender(String gender);  
  
}
```

```
package com.javabykiran;  
  
abstract class AbstclassEx3 extends AbstractEx3 {  
    abstract void getCity(String city);  
  
    abstract void getCountry(String country);  
  
}
```

```
package com.javabykiran;  
  
class AbstractSubclassEx3 extends AbstclassEx3 {  
    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 : " + gender);  
    System.out.println("City : " + city);  
    System.out.println("Country : " + country);  
}  
}
```

```
package com.javabykiran;  
  
public class AbstractTestEx3 {  
    public static void main(String args[]) {  
  
        AbstractSubclassEx3 obj = new AbstractSubclassEx3();  
        obj.getName("Javabykiran");  
        obj.getGender("rather not say");  
        obj.getCity("Pune");  
        obj.getCountry("India");  
        obj.display();  
    }  
}
```

- Access Abstract Class Methods within a Package in Java

```
package com.javabykiran;  
  
public abstract class AbstractEx4 {  
  
    abstract void getVal(String name);  
  
    abstract void display();  
}
```

```
package com.javabykiran;

public class AbstractSubclassEx4 extends AbstractEx4 {
    String name;

    void getVal(String name) {
        this.name = name;
    }

    void display() {
        System.out.println("Hello " + name);
    }
}
```

```
package com.javabykiran;

class AbstractTestEx4 {

    public static void main(String args[]) {
        AbstractSubclassEx4 obj = new AbstractSubclassEx4();
        obj.getVal("SampleCodez");
        obj.display();
    }
}
```

- Simple example of interface

```
package com.javabykiran;

public interface Interface11 {
    public void myMethod();
}

package com.javabykiran;

public interface Interface22 {
    public void myMethod();
}
```

```
package com.javabykiran;

public class InterfaceTestEx1 implements Interface11, Interface22 {
    public void myMethod() {
        System.out.println(" Multiple inheritance example using interfaces");
    }

    public static void main(String[] args) {
        InterfaceTestEx1 id = new InterfaceTestEx1();
        id.myMethod();
    }
}
```

- Simple example on inherited interface

```
package com.javabykiran;

public interface InterfaceEx21 {

    public void method1();
}
```

```
package com.javabykiran;

public interface InterfaceEx22 extends InterfaceEx21 {
    public void method2();
}
```

```
package com.javabykiran;

public class InterfaceTestEx2 implements InterfaceEx22 {
    public void method1() {
        System.out.println("Implemented method1");
    }
}
```

```
public void method2() {  
    System.out.println("Implemented method2");  
}  
  
public static void main(String[] args) {  
    InterfaceTestEx2 t = new InterfaceTestEx2();  
    t.method1();  
    t.method1();  
}  
}
```

- Example of interface with variable

```
package com.javabykiran;  
  
public interface Moveable {  
    int AVGSPEED = 40;  
  
    void move();  
}
```

```
package com.javabykiran;  
  
class Vehicle implements Moveable {  
    public void move() {  
        System.out.println("Average speed is" + AVGSPEED);  
    }  
  
    public static void main(String[] arg) {  
        Vehicle vc = new Vehicle();  
        vc.move();  
    }  
}
```

- Interface example

```
package com.javabykiran;  
  
public interface MovableEx2 {  
  
    boolean isMoveable();  
}
```

```
package com.javabykiran;  
  
public interface Rollable {  
  
    boolean isRollable();  
}
```

```
package com.javabykiran;  
  
public class Tyre implements MovableEx2, Rollable {  
    int width;  
  
    public boolean isMoveable() {  
        return true;  
    }  
  
    public boolean isRollable() {  
        return true;  
    }  
  
    public static void main(String args[]) {  
        Tyre tr = new Tyre();  
        System.out.println(tr.isMoveable());  
        System.out.println(tr.isRollable());  
    }  
}
```


- **Interface example**

```
package com.javabykiran;  
  
public interface InterfaceEx4 {  
    public void method1();  
  
    public void method2();  
}
```

Homework

- Solve test on jbktest.com for abstraction
- Read jbktutorials.com
 - <https://www.jbktutorials.com/corejava/abstraction-in-java.php#gsc.tab=0>
- Read interview questions
 - <https://www.jbktutorials.com/core-java-interview-questions/abstraction-interview-questions.php#gsc.tab=0>

Download

- <https://drive.google.com/drive/folders/1-O3m9ira-mbYJGZYgDXkiuwSiNUCL5Li?usp=sharing>