

10. INTERFACES

1.

```
interface MyInterface
{
    public void method1();
}
class Main implements MyInterface
{
    public void method1()
    {
        System.out.println("method1...");
    }
    public static void main(String arg[])
    {
        MyInterface obj = new Main();
        obj.method1();
    }
}
```

Output-
method1...

2.

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

Output-
method1...

3.

We can't create instance(interface can't be instantiated) of interface but we can make reference of it that refers to the Object of its implementing class.

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

Output-
method1...
method2...

4.

```
interface MyInterface1
{
    public void method1();
}
interface MyInterface2
{
    public void method2();
}
```

```
class Main implements MyInterface1,MyInterface2
{
    public void method1()
    {
        System.out.println("method1...");
    }
    public void method2()
    {
        System.out.println("method2...");
    }
}
```

```

    }
    public static void main(String arg[])
    {
        MyInterface1 obj1 = new Main();
        obj1.method1();
        MyInterface2 obj2 = new Main();
        obj2.method2();
    }
}

```

Output-
method1...
method2...

5.

```

interface MyInterface1
{
    public void method();
}
interface MyInterface2
{
    public void method();
}

```

class Main implements MyInterface1,MyInterface2

```

{
    public void method()
    {
        System.out.println("method...");
    }
    public static void main(String arg[])
    {
        MyInterface1 obj1 = new Main();
        obj1.method();
        MyInterface2 obj2 = new Main();
        obj2.method();
    }
}

```

Output-
method...
method...

6.

```

interface MyInterface
{
    default void method()
    {
        System.out.println("method...");
    }
}

```

```

    }
}
class Main implements MyInterface
{
    public static void main(String arg[])
    {
        MyInterface obj = new Main();
        obj.method();
    }
}

```

Output-
method...

```

7.
import java.io.*;
interface intA
{
    void method1();
}

interface intB extends intA
{
    void method2();
}

class Main implements intB
{
    public void method1()
    {
        System.out.println("method1...");
    }
    public void method2()
    {
        System.out.println("method2...");
    }
}

public static void main (String[] args)
{
    Main ob1 = new Main();
    ob1.method1();
    ob1.method2();
}
}

```

Output-
method1...
method2...

8.

```

interface MyInterface
{
    public static int num = 10;
    public void display();
}
public class Main implements MyInterface
{
    public static int num = 1000;
    public void display()
    {
        System.out.println("This is the implementation of the display method");
    }
    public void show()
    {
        System.out.println("This is the implementation of the show method");
    }
    public static void main(String args[]) {
        Main obj = new Main();
        System.out.println("Value of num of the interface "+MyInterface.num);
        System.out.println("Value of num of the class "+obj.num);
    }
}

```

Output-

Value of num of the interface 10
 Value of num of the class 1000

9.

```

public interface MyInterface
{

    public abstract void mul(int a, int b);
    public default void
    add(int a, int b)
    {
        sub(a, b);
        div(a, b);
        System.out.print("Default method = ");
        System.out.println(a + b);
    }

    public static void mod(int a, int b)
    {
        div(a, b);
        System.out.print("Static method = ");
        System.out.println(a % b);
    }

    private void sub(int a, int b)

```

```

    {
        System.out.print("Private method = ");
        System.out.println(a - b);
    }

    private static void div(int a, int b)
    {
        System.out.print("Private static method = ");
        System.out.println(a / b);
    }
}
class Main implements MyInterface
{
    public void mul(int a, int b)
    {
        System.out.print("Abstract method = ");
        System.out.println(a * b);
    }

    public static void main(String[] args)
    {
        MyInterface obj = new Main();
        obj.mul(2, 3);
        obj.add(6, 2);
        MyInterface .mod(5, 3);
    }
}

```

Output-

```

Abstract method = 6      // mul(2, 3) = 2*3 = 6
Private method = 4      // sub(6, 2) = 6-2 = 4
Private static method = 3 // div(6, 2) = 6/2 = 3
Default method = 8      // add(6, 2) = 6+2 = 8
Private static method = 1 // div(5, 3) = 5/3 = 1
Static method = 2       // mod(5, 3) = 5%3 = 2

```

10.

If the fields of the interface are private/protected, we cannot access them in the implementing class. If we try to declare the fields of an interface private, a compile time error is generated saying "modifier private not allowed here" or "modifier protected not allowed here".

```

interface MyInterface
{
    public static int num1 = 10;
    private static final int num2 = 10;
    protected static final int num3 = 10;
    public void display();
}
public class Main implements MyInterface
{

```

```

public static int num = 1000;
public void display()
{
    System.out.println("This is the implementation of the display method");
}
public void show()
{
    System.out.println("This is the implementation of the show method");
}
public static void main(String args[]) {
    Main obj = new Main();
    System.out.println("Value of num of the interface "+MyInterface.num1);
    System.out.println("Value of num of the class "+obj.num1);
    System.out.println("Value of num of the interface "+MyInterface.num2);
    System.out.println("Value of num of the class "+obj.num2);
    System.out.println("Value of num of the interface "+MyInterface.num3);
    System.out.println("Value of num of the class "+obj.num3);
}
}

```

Output-

Main.java:4: error: modifier private not allowed here

```
private static final int num2 = 10;
```

^

Main.java:5: error: modifier protected not allowed here

```
protected static final int num3 = 10;
```

^

2 errors

11.

interface MyInterface

```

{
    static final int num = 10;
    public void display();
}

```

public class Main implements MyInterface

```

{
    public static int num = 1000;
    public void display()
    {
        System.out.println("This is the implementation of the display method");
    }
    public void show()
    {
        System.out.println("This is the implementation of the show method");
    }
    public static void main(String args[]) {
        Main obj = new Main();
        System.out.println("Value of num of the interface "+MyInterface.num);
        System.out.println("Value of num of the class "+obj.num);
    }
}

```

```
}  
}
```

Output-

Value of num of the interface 10

Value of num of the class 1000