Interfaces

By

Dr M. Senthilkumar

Assistant Professor

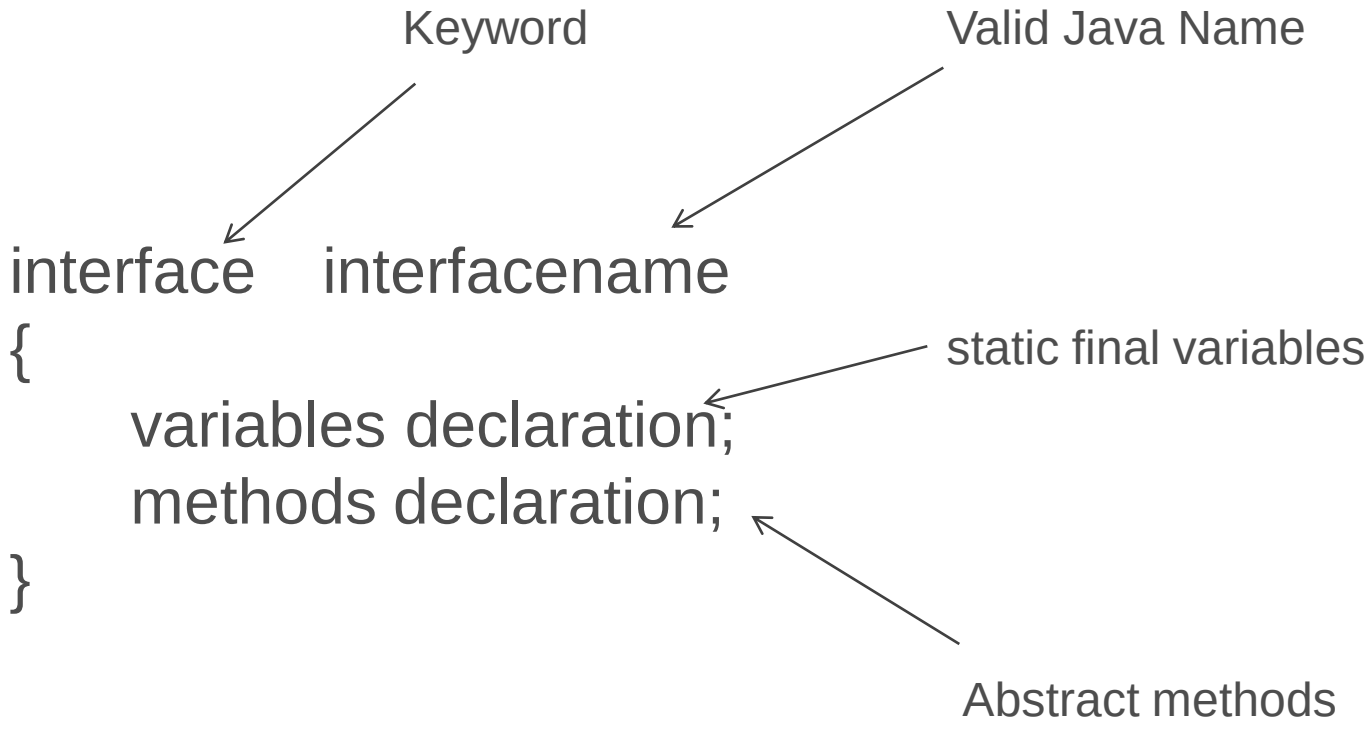
Department of Computer Science

Government Arts and Science College, Avinashi - 641654

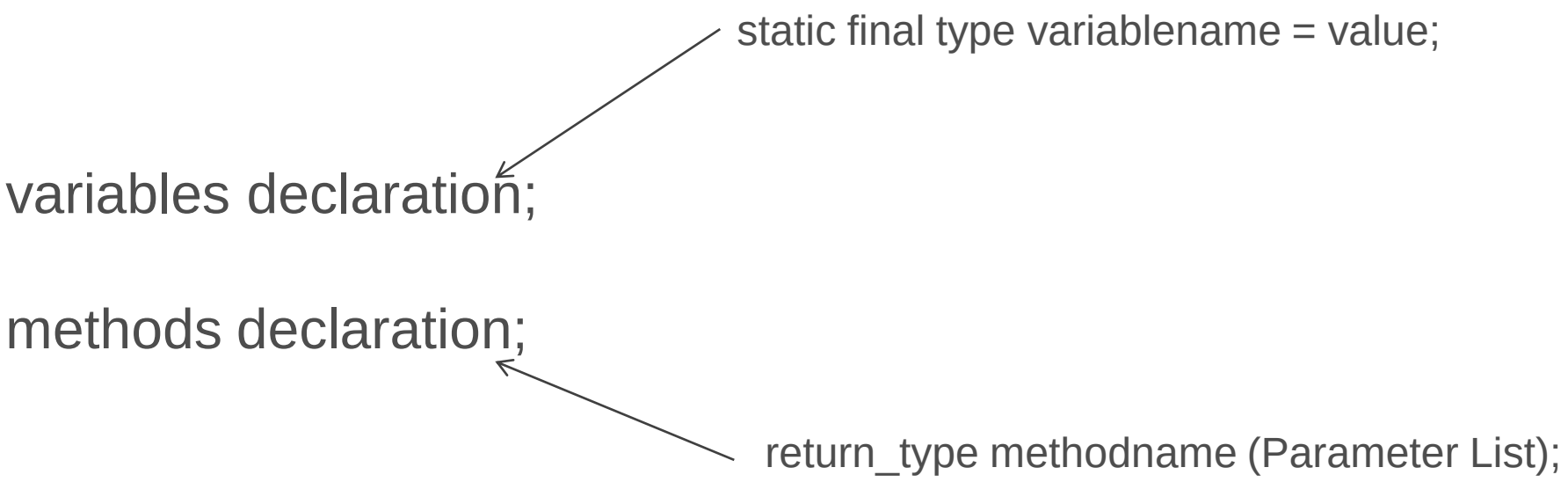
Interface

- ✓ Similar to a Class
- ✓ Contains abstract methods and final fields
- ✓ A class that implements the Interface must define the code for abstract methods

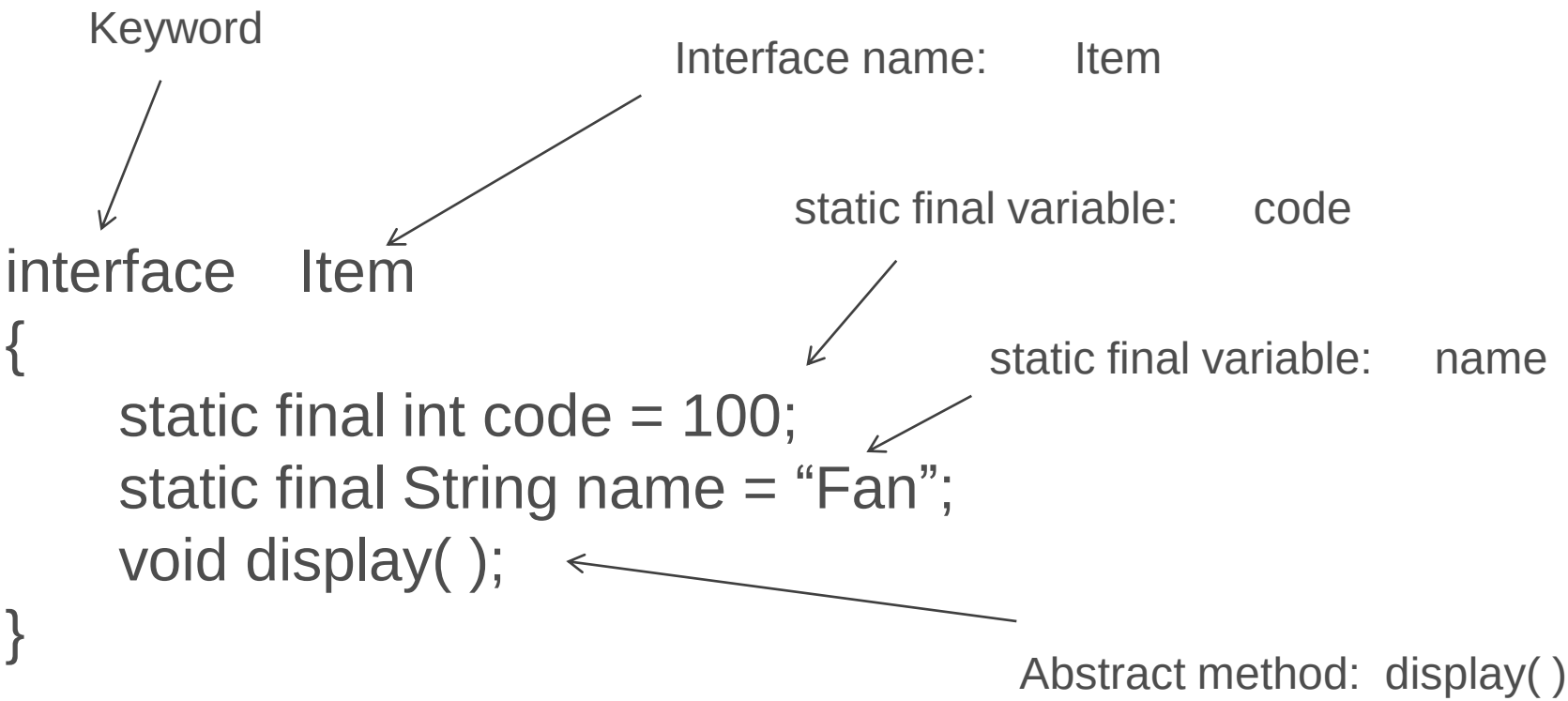
Syntax


Keyword
Valid Java Name
interface interfacename
{
variables declaration;
methods declaration;
}
static final variables
Abstract methods

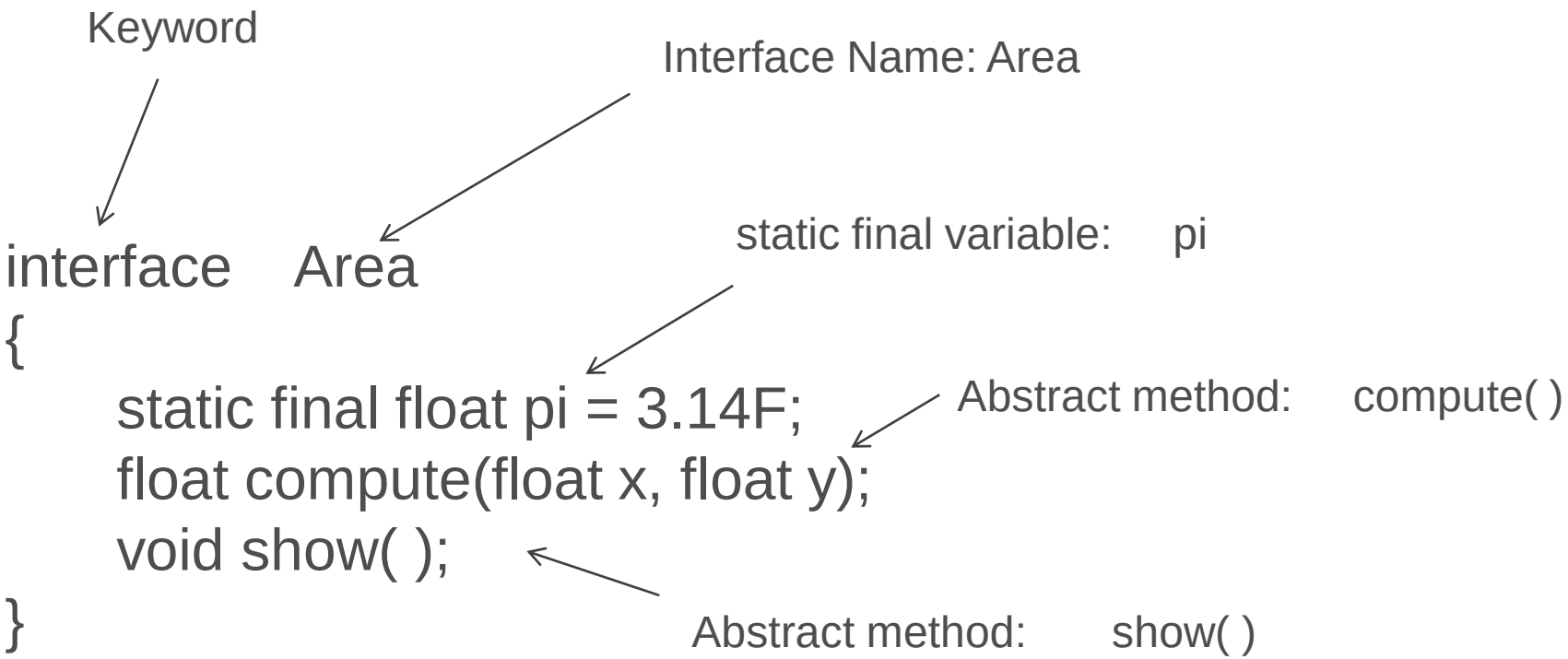
Syntax


static final type variablename = value;
variables declaration;
methods declaration;
return_type methodname (Parameter List);

Example 1


Keyword
Interface name: Item
static final variable: code
static final variable: name
interface Item
{
static final int code = 100;
static final String name = “Fan”;
void display( );
}
Abstract method: display( )

Example 2


Keyword
Interface Name: Area
interface Area
{
static final float pi = 3.14F;
float compute(float x, float y);
void show( );
}
static final variable: pi
Abstract method: compute( )
Abstract method: show( )

Extending Interfaces

- ✓ An interface can extend one or many interfaces
- ✓ Inheritance property

```
interface    interface2  extends interface1
{
        body of interface2;
}
```

Example 3

```
interface  ItemConstants
{
        int itemcode = 100;
        String itemname = “Fan”;
        float unitprice = 1500;
        int quantity = 10;
}
interface  ItemMethods
{
        void computecost( );
}
interface  Item extends ItemConstants, ItemMethods
{
        void display( );
}
```

Implementing Interfaces

- ✓ A class can inherit properties of one or more Interfaces
- ✓ A class can implement one or more Interfaces

```
class classname implements interfacename
{
        body of the classname
}
```

Implementing Interfaces

- ✓ A class can inherit properties of one or more
  - ✓ Classes (use extends)
  - ✓ Interfaces (use implements)

```
class classname extends superclass1, superclass2
implements interfacename1, interfacename2
{
        body of the classname
}
```

Example 4

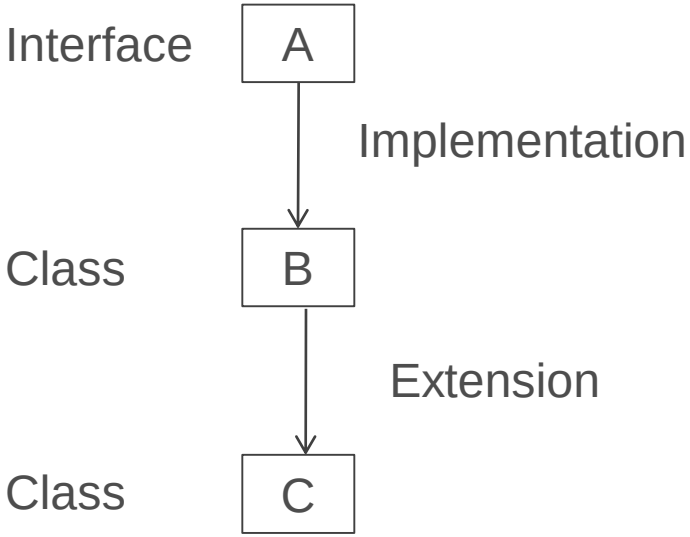
```
interface  Area
{
      final static float pi = 3.14F;
      float compute(float x, float y);
}
class Rectangle implements Area
{
      public float compute(float x, float y)
      {
            return(x * y);
      }
}
class Circle implements Area
{
      public float compute(float x, float y)
      {
            return(pi * x * x);
      }
}
```

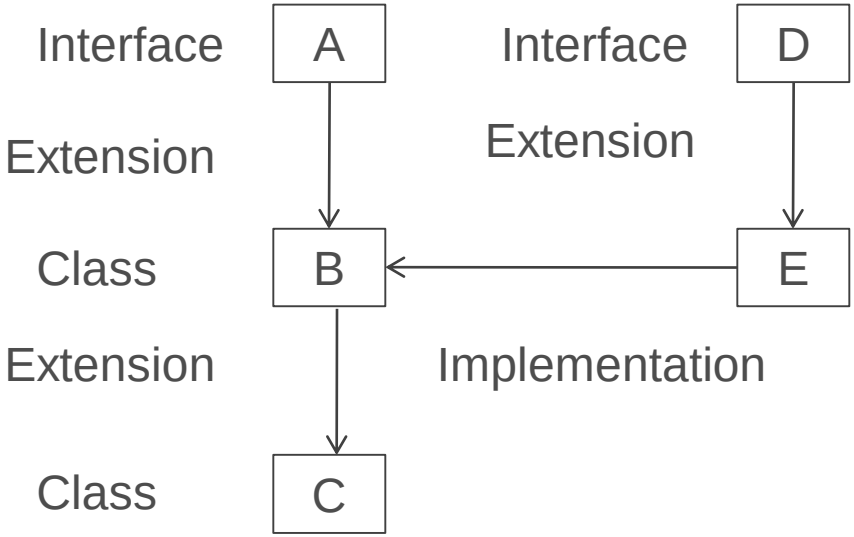
```
class InterfaceExample2
{
public static void main(String args[])
  {
      Rectangle R = new Rectangle( );
      Circle C = new Circle( );
      System.out.println("Rectangle Area: " + R.compute(10,20));
      System.out.println("Circle Area: " + C.compute(10,0));
  }
}
```

Example 4 - Output

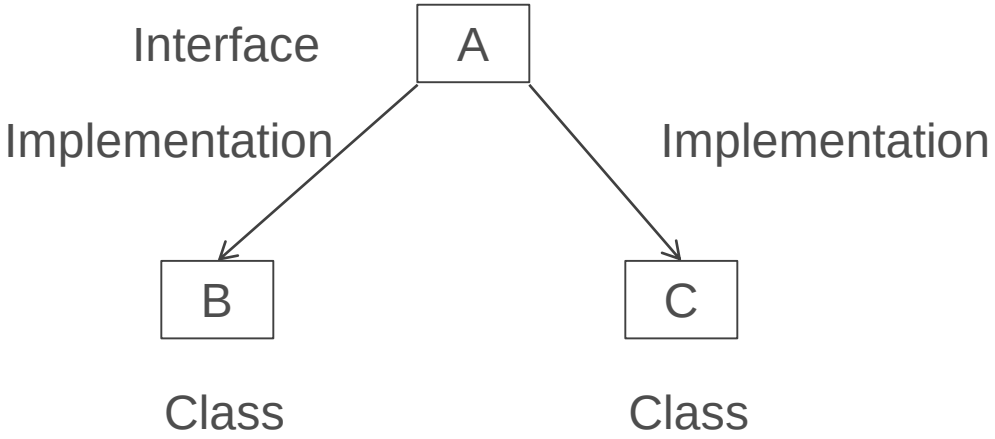
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Senthil>d:

D:\>cd D:\jdk1.8.0_111\jdk1.8.0_111\bin

D:\jdk1.8.0_111\jdk1.8.0_111\bin>javac InterfaceExample2.java

D:\jdk1.8.0_111\jdk1.8.0_111\bin>java InterfaceExample2
Rectangle Area: 200.0
Circle Area: 314.0

D:\jdk1.8.0_111\jdk1.8.0_111\bin>
```

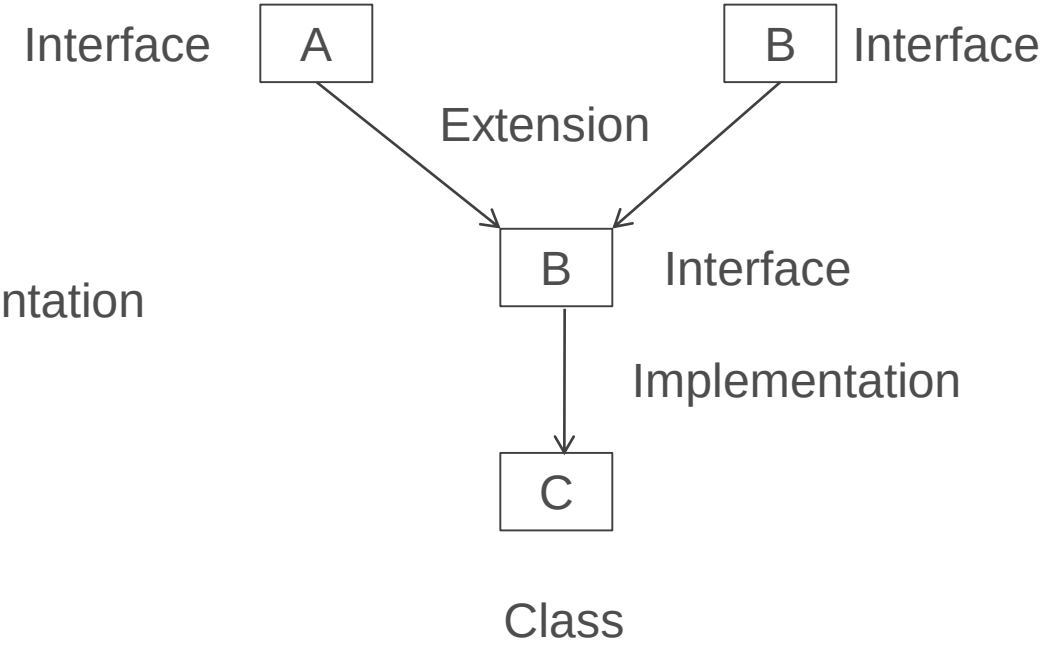
Forms of Interface Implementation





Forms of Interface Implementation





Example 5

```
class Student
{
      int rollNo;
      void getNumber(int   n)
      {
         rollNo = n;
      }
      void putNumber( )
      {
         System.out.println("Roll No :  " + rollNo);
      }
}
class Test  extends Student
{
      float  JavaMarks, DSMarks;
      void getMarks(float  m1, float  m2)
      {
          JavaMarks = m1; DSMarks = m2;
      }
      void putMarks( )
      {
         System.out.println("Marks Received : ");
         System.out.println("Java Marks:  " + JavaMarks);
         System.out.println("DS Marks:  " + DSMarks);
      }
}


interface sports
{
      float SportsMarks = 6.0F;
      void putSportsMarks( );
}
```

Example 5

```
class Results extends Test implements Sports
{
   float   total;
   public void putSportsMarks( )
   {
      System.out.println("Sports Marks:  "+ SportsMarks);
   }
   void display( )
   {
      total = JavaMarks + DSMarks + SportsMarks;
      putNumber( );
      putMarks( );
      putSportsMarks( );
      System.out.println("Total Score:  "+ total);
   }
}
```

```
class HybridInterface
{
   public static void main(String  args[ ])
   {
      Results S1 = new Results( );
      S1.getNumber(100);
      S1.getMarks(27.5F, 33.0F);
      S1.display( );
   }
}
```

Example 5 - Output

```
D:\jdk1.8.0_111\jdk1.8.0_111\bin>javac HybridInterface.java

D:\jdk1.8.0_111\jdk1.8.0_111\bin>java HybridInterface
Roll No :  100
Marks Received :
Java Marks:  27.5
DS Marks:  33.0
Sports Marks:  6.0
Total Score:  66.5

D:\jdk1.8.0_111\jdk1.8.0_111\bin>
```

References

- ✓ Programming with Java – A Primer - E. Balagurusamy, 3rd Edition, TMH

Thank You