

PROGRAM – 1

```
class Player{

    String name;

    int age;

    Player(String n,int a){

        name=n;

        age=a;

    }

    void show()

    {

        System.out.println("Player name: "+name);

        System.out.println("Age: "+age);

    }

}

class cricket_player extends Player{

    String type;

    cricket_player(String n,String t,int a){

        super(n,a);

        type=t;

    }

    public void show()

    {

        super.show();

        System.out.println("Player type : "+type);

    }

}
```

```
class football_player extends Player{  
    String type;  
    football_player(String n,String t,int a)  
{  
    super(n,a);  
    type=t;  
}  
    public void show()  
{  
    super.show();  
    System.out.println("Player type : "+type);  
}  
}
```

```
class hockey_player extends Player{  
    String type;  
    hockey_player(String n,String t,int a)  
{  
    super(n,a);  
    type=t;  
}  
    public void show()  
{  
    super.show();  
    System.out.println("Player type : "+type);  
}  
}
```

```

    }

    public class Main{

        public static void main(String args[])

        {

            cricket_player c=new cricket_player("Rohit","Cricket",20);

            football_player f=new football_player("Rahul","Football",21);

            hockey_player h=new hockey_player("Aditya","Hockey",22);

            c.show();

            f.show();

            h.show();

        }

    }

```



The screenshot shows a Java IDE with a dark theme. The top pane displays the source code for a `Player` class and a `Main` class. The `Player` class has attributes `String name` and `int age`, a constructor `Player(String n, int a)`, and a `show()` method. The `Main` class has a `main` method that creates three objects: `cricket_player` (Rohit, 20), `football_player` (Rahul, 21), and `hockey_player` (Aditya, 22), and calls their `show()` methods. The bottom pane shows the output of the program, which prints the details of each player. The output is as follows:

```

Player name: Rohit
Age: 20
Player type : Cricket
Player name: Rahul
Age: 21
Player type : Football
Player name: Aditya
Age: 22
Player type : Hockey

...Program finished with exit code 0
Press ENTER to exit console.

```

PROGRAM - 2

```
class Rectangle{

    int length, breadth;

    Rectangle(int l, int b){

        length = l;

        breadth = b;

    }

    void area(){

        System.out.println("Area of Rectangle: "+(length*breadth));

    }

    void perimeter(){

        System.out.println("Perimeter of Rectangle: "+ (2*(length+breadth)));

    }

}

class Square extends Rectangle{

    int side;

    Square(int s, int l, int b){

        super(l, b);

        side = s;

    }

    public void area(){

        super.area();

        System.out.println("Area of Square: "+(side*side));

    }

    public void perimeter(){

        super.perimeter();

        System.out.println("Perimeter of Square: "+(4*side));

    }

}
```

```
}  
}
```

```
public class Main
```

```
{  
  
    public static void main(String[] args) {  
  
        Square sq = new Square(8, 20, 15);  
  
        sq.area();  
  
        sq.perimeter();  
  
    }  
}
```

```
1- class Rectangle{  
2-     int length, breadth;  
3-     Rectangle(int l, int b){  
4-         length = l;  
5-         breadth = b;  
6-     }  
7-  
8-     void area(){  
9-         System.out.println("Area of Rectangle: "+(length*breadth));  
10-    }  
11-  
12-    void perimeter(){  
13-        System.out.println("Perimeter of Rectangle: "+ (2*(length+breadth)));  
14-    }  
15- }  
16-  
17- class Square extends Rectangle{  
18-     int side;  
19-  
20-     void area(){  
21-         System.out.println("Area of Square: "+(side*side));  
22-     }  
23-  
24-     void perimeter(){  
25-         System.out.println("Perimeter of Square: "+ (2*side));  
26-     }  
27- }  
28- }
```

input

```
Area of Rectangle: 300  
Area of Square: 64  
Perimeter of Rectangle: 70  
Perimeter of Square: 32  
  
...Program finished with exit code 0  
Press ENTER to exit console.[]
```

PROGRAM – 3

```
interface Printable // creating Interface 1
{
    void print();
}

interface Showable // creating Interface 2{
void show();
}

class Demo implements Printable,Showable{
    public void print()
    {
        System.out.println("Welcome");
    }

    public void show()
    {
        System.out.println("Interface Demo");
    }
}

public class Main{
    public static void main(String args[])
    {
        Demo obj = new Demo();

        obj.print();

        obj.show();
    }
}
```

```
9 void show();
10 }
11
12 class Demo implements Printable, Showable
13 {
14     public void print()
15     {
16         System.out.println("Welcome");
17     }
18     public void show()
19     {
20         System.out.println("Interface Demo");
21     }
22 }
23
24 public class Main{
25     public static void main(String args[])
26     {
27         Demo obj = new Demo();
28     }
29 }
```

input

Welcome
Interface Demo

...Program finished with exit code 0
Press ENTER to exit console.

PROGRAM – 4

interface Animal // creating Interface 1

```
{
    int a=10; // implicitly static and final
    void eat(); //methods of interface are implicitly public
    void sleep();
}
```

class Demo implements Animal

```
{
    public void eat()
    {
        System.out.println("Animal Eats!");
    }
}
```

public void sleep()

```
{
```

```

        System.out.println("Animal Sleeps!");

        System.out.println("Static variable: "+a);
    }
}

```

```

public class Main{

    public static void main(String args[])

    {

        Demo obj = new Demo();

        obj.eat();

        obj.sleep();

    }

}

```

The screenshot shows a Java IDE with a file named 'Main.java'. The code defines an interface 'Animal' with a static variable 'a' and two methods 'eat()' and 'sleep()'. A class 'Demo' implements these methods. The 'eat()' method prints 'Animal Eats!'. The 'sleep()' method prints 'Animal Sleeps!' and 'Static variable: '+a'. The output window shows the execution results: 'Animal Eats!', 'Animal Sleeps!', and 'Static variable: 10'. The program finished with exit code 0.

```

Main.java
1 interface Animal // creating Interface 1
2 {
3     int a=10; // implicitly static and final
4     void eat(); //methods of interface are implicitly public
5     void sleep();
6 }
7
8 class Demo implements Animal
9 {
10     public void eat()
11     {
12         System.out.println("Animal Eats!");
13     }
14
15     public void sleep()
16     {
17         System.out.println("Animal Sleeps!");
18         System.out.println("Static variable: "+a);
19     }
20 }

```

input

```

Animal Eats!
Animal Sleeps!
Static variable: 10

...Program finished with exit code 0
Press ENTER to exit console.

```