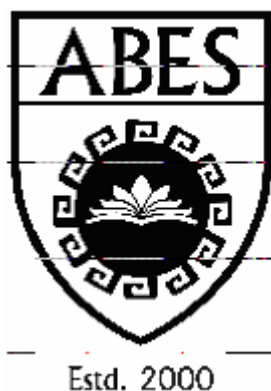


# Practical File

**Lab Name**..... **Lab Code**.....



**Name**.....

**Adm.No**..... **Univ. Roll No**.....

**Course** ..... **Branch**.....

**Sem**..... **Section**.....

**ABES**  
**Engineering College**  
(College Code-032)

NAAC Accredited, NBA Accredited Branches (CSE, ECE, EN & IT)

19th Km. Stone, NH-09, Ghaziabad - 201009 (UP), India

Phone : 0120-7135112, 9999889341 Fax : 0120-7135115 Website : [www.abes.ac.in](http://www.abes.ac.in), Email: [info@abes.ac.in](mailto:info@abes.ac.in)



**1. PRACTICAL STATEMENT OF PRACTICAL:**

Write a program to implement Multiple Classes in Java.

**2. IMPLEMENTATION**

```
class Computer {  
    Computer() {  
        System.out.println("Constructor of Computer class.");  
    }  
  
    void computer_method() {  
        System.out.println("Power gone! Shut down your PC soon...");  
    }  
  
    public static void main(String[] args) {  
        Computer my = new Computer();  
        Laptop your = new Laptop();  
  
        my.computer_method();  
        your.laptop_method();  
    }  
}  
class Laptop {  
    Laptop() {  
        System.out.println("Constructor of Laptop class.");  
    }  
  
    void laptop_method() {  
        System.out.println("99% Battery available.");  
    }  
}
```

### 3. Result /Output

```
java -cp /tmp/kSiomjTuQH Computer  
Constructor of Computer class.  
Constructor of Laptop class.  
Power gone! Shut down your PC soon  
99% Battery available.
```

## 1. PRACTICAL STATEMENT OF PRACTICAL:

Write a program to implement Constructor Overloading in Java.

## 2. IMPLEMENTATION

```
class StudentData
{
    private int stuID;
    private String stuName;
    private int stuAge;
    StudentData()
    {
        //Default constructor
        stuID = 100;
        stuName = "New Student";
        stuAge = 18;
    }
    StudentData(int num1, String str, int num2)
    {
        //Parameterized constructor
        stuID = num1;
        stuName = str;
        stuAge = num2;
    }
    //Getter and setter methods
    public int getStuID() {
        return stuID;
    }
    public void setStuID(int stuID) {
        this.stuID = stuID;
    }
    public String getStuName() {
        return stuName;
    }
    public void setStuName(String stuName) {
        this.stuName = stuName;
    }
}
```

```

public int getStuAge() {
    return stuAge;
}
public void setStuAge(int stuAge) {
    this.stuAge = stuAge;
}

public static void main(String args[])
{
    //This object creation would call the default constructor
    StudentData myobj = new StudentData();
    System.out.println("Student Name is: "+myobj.getStuName());
    System.out.println("Student Age is: "+myobj.getStuAge());
    System.out.println("Student ID is: "+myobj.getStuID());

    /*This object creation would call the parameterized
    * constructor StudentData(int, String, int)*/
    StudentData myobj2 = new StudentData(555, "Chaitanya", 25);
    System.out.println("Student Name is: "+myobj2.getStuName());
    System.out.println("Student Age is: "+myobj2.getStuAge());
    System.out.println("Student ID is: "+myobj2.getStuID());
}
}

```

### 3. Result/Output

```
java -cp /tmp/y6LokwjLb1 StudentData
```

tangular Snip

```

Student Name is: New StudentStudent Age is: 18
Student ID is: 100Student Name is: Chaitanya
Student Age is: 25
Student ID is: 555

```

## 1. PRACTICAL STATEMENT OF PRACTICAL:

Write a program to implement Method Overloading in Java.

## 2. IMPLEMENTATION

```
class DisplayOverloading
{
    public void disp(char c)
    {
        System.out.println(c);
    }
    public void disp(char c, int num)
    {
        System.out.println(c + " "+num);
    }
}
class Sample
{
    public static void main(String args[])
    {
        DisplayOverloading obj = new DisplayOverloading();
        obj.disp('a');
        obj.disp('a',10);
    }
}
```

### 3. Result/Output

## Output

```
java -cp /tmp/y6LokwjLb1 Sample
```

```
.tangular Snip
```

```
a
```

```
a 10
```

```
|
```

### 1. PRACTICAL STATEMENT OF PRACTICAL:

Write a program to implement Single Level Inheritance in Java.

### 2. IMPLEMENTATION

```
class Employee
{
    float sal=60000;
}
class Main extends Employee
{
    float b=1500;
    float temp= sal + b;
    public static void main(String args[])
    {
        Main ob=new Main();
        System.out.println("Salary amount is:"+ob.sal);
        System.out.println(" Extra Bonous is:"+ob.temp);
    }
}
```



### 3. Result/Output

```
Salary amount is:60000.0  
Extra Bonous is:61500.0
```

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

**1. PRACTICAL STATEMENT OF PRACTICAL:**

Write a program to implement Multiple Level Inheritance in Java.

**2. IMPLEMENTATION**

```
class Person
{
    Person()
    {
        System.out.println("Person constructor");
    }
    void nationality()
    {
        System.out.println("Indian");
    }
    void place()
    {
        System.out.println("Mumbai");
    }
}
```

```
class Emp extends Person
{
Emp()
{
System.out.println("Emp constructor");
}
void organization()
{
System.out.println("IBM");
}
void place()
{
System.out.println("New York");
}
}
```

```
class Manager extends Emp
{
    Manager()
    {
        System.out.println("Manager constructor");
    }
    void subordinates()
    {
        System.out.println(12);
    }
    void place()
    {
        System.out.println("London");
    }
}
```

```
class Check
{
    public static void main(String arg[])
    {
        Manager m=new Manager();
        m.nationality();
        m.organization();
        m.subordinates();
        m.place();
    }
}
```

### 3. Result/Output

#### Output

```
java -cp /tmp/y6LokwjLb1 Check
```

```
Person constructorEmp constructor
```

```
Manager constructor
```

```
Indian
```

```
IBM
```

```
12
```

```
London
```

```
|
```

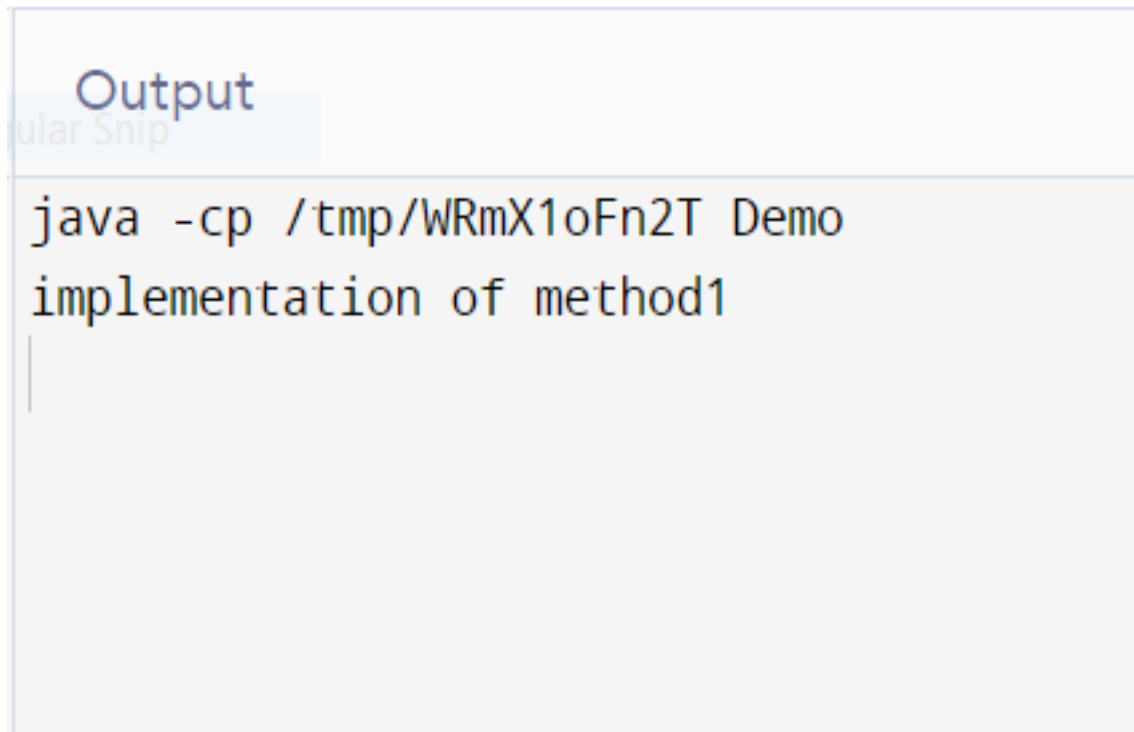
**1. PRACTICAL STATEMENT OF PRACTICAL:**

Write a program to implement Interface in Java.

**2. IMPLEMENTATION**

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

### 3. Result/Output

A screenshot of a terminal window. The title bar at the top says "Output" and "ular Snip". The terminal content shows a command being executed: "java -cp /tmp/WRmX1oFn2T Demo implementation of method1". The command is split across two lines. The first line is "java -cp /tmp/WRmX1oFn2T Demo" and the second line is "implementation of method1". There is a vertical cursor line at the end of the second line.

```
Output
ular Snip

java -cp /tmp/WRmX1oFn2T Demo
implementation of method1
|
```

**1. PRACTICAL STATEMENT OF PRACTICAL:**

Use basic tag in HTML and XHTML

**2. IMPLEMENTATION**

**3. Result/Output**



## 1. PRACTICAL STATEMENT OF PRACTICAL:

Write a program to use links, tables, images and videos

## 2. IMPLEMENTATION

```
<html>
<head>
<style>

    table, th, td {
        border: 1px solid black;
    }

</style>
</head>
<body>
    <table>
        <tr>
            <th>Name</th>
            <th>Age</th>
        </tr>
        <tr>
            <td>VIjay</td>
            <td>20</td>
        </tr>
        <tr>
            <td>Kumar</td>
            <td>21</td>
        </tr>
    </table>
</body>
</html>
```

```
<html>
<head>
    <title>Link</title>
</head>
<body>
    <a href="https://www.c-sharpcorner.com/">Visit My blog</a>
</body>
</html>
```

### 3. Result/Output

---

Name	Age
Vijay	20
Kumar	21

[Visit My blog](#)

## 1. PRACTICAL STATEMENT OF PRACTICAL:

Write a program to create menu using HTML and CSS

## 2. IMPLEMENTATION

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}

.topnav {
  overflow: hidden;
  background-color: #333;
}

.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```

.topnav a.active {
  background-color: #04AA6D;
  color: white;
}
</style>
</head>
<body>

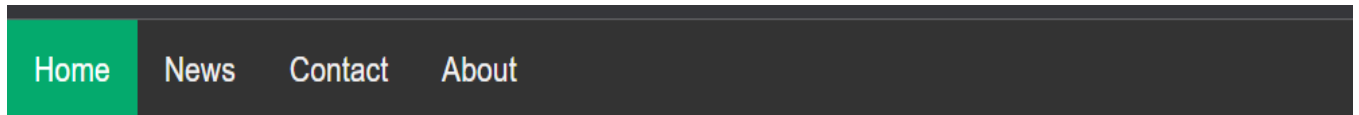
<div class="topnav">
  <a class="active" href="#home">Home</a>
  <a href="#news">News</a>
  <a href="#contact">Contact</a>
  <a href="#about">About</a>
</div>

<div style="padding-left:16px">
  <h2>Top Navigation Example</h2>
  <p>Some content..</p>
</div>

</body>
</html>

```

### 3. Result/Output



### 1. PRACTICAL STATEMENT OF PRACTICAL:

Write a program to show the usage of frame in HTML

### 2. IMPLEMENTATION

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset rows = "10%,80%,10%">
    <frame name = "top" src = "/html/top_frame.htm" />
    <frame name = "main" src = "/html/main_frame.htm" />
    <frame name = "bottom" src = "/html/bottom_frame.htm" />
  </frameset>

  <noframes>
    <body>Your browser does not support frames.</body>
  </noframes>

</html>
```

### 3. Result/Output

Practical Name..... Practical No.....

Cannot GET /html/top\_frame.htm

---

Cannot GET /html/main\_frame.htm