



MCA Sem - I

Lab Manual

Subject: 1MCA8: OOP-JAVA PROGRAMMING

LAB MANUAL

City College

**No. 27/2, 33rd Cross, 2nd Main Rd, 7th Block,
Jayanagar, Bengaluru,
Karnataka 560070**

1. Develop a JAVA program to demonstrate the precedence and associativity among arithmetic operators. The program should also demonstrate how the default precedence can be overridden.

```
class Prg1
{
    public static void main(String args[])
    {
        float i = 40;
        float j = 80;
        float k = 40;

        float l = i + j / k; //precedence of division higher than
addition
        System.out.println("value of L: " + l);

        System.out.println("Precedence overriding: Using brackets");
        l = (i + j) / k; // overriding default precedence using brackets
        System.out.println("Value of L: " + l);

        float n = i / k * 2 + j; //Associativity- division first then
multiplication.
        System.out.println("\n\nvalue of n: " + n);

        System.out.println("Overriding associativity: Using brackets");
        n = i / (k * 2) + j;// overriding associativity using brackets
        System.out.println("Value of n: " + n);
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mandal\Downloads\Java Lab 2023\java Prg1.java
value of L: 42.0
Precedence overriding: Using brackets
Value of L: 3.8

value of n: 88.9
Overriding associativity: Using brackets
Value of n: 88.9

C:\Users\mandal\Downloads\Java Lab 2023>
```

2. Write a JAVA program to validate a date. The program should accept day, month and year and it should report whether they form a valid date or not.

```
import java.util.Scanner;

class Prg2
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int day=0, month=0, year=0;
        try
        {
            System.out.println("Enter Day: ");
            day = sc.nextInt();
            System.out.println("Enter Month");
            month = sc.nextInt();
            System.out.println("Enter Year");
            year = sc.nextInt();
        }
        catch (Exception e)
        {
            System.out.println("Input Error—InValid Data—Re-Enter.. "+e);
            System.exit(0);
        }
        System.out.print("\n"+day);
        System.out.print("/");
    }
}
```

```
System.out.print(+month);
System.out.print("/");
System.out.print(+year + " ");

boolean isTrueDate = true;

if ((day<=0) || (month<=0) || (year<=0))
{
    isTrueDate = false;
}
else if(month > 12)
{
    isTrueDate = false;
}
else if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 ||
month == 10 || month == 12)
{
    if (day <= 31)
    {
        isTrueDate = true;
    }
    else
    {
        isTrueDate = false;
    }
}
else if (month == 4 || month == 6 || month == 9 || month == 11)
{
    if (day <= 30)
    {
        isTrueDate = true;
    }
    else
    {
        isTrueDate = false;
    }
}

else if (month == 2) // February check
{
    if (year % 4 == 0) // Leap year check for February
    {
        if (day <= 29)
        {
            isTrueDate = true;
        }
        else
        {
            isTrueDate = false;
        }
    }
}
```

```

        else if (year % 4 != 0)
        {
            if (day <= 28)
            {
                isTrueDate = true;
            }
            else
            {
                isTrueDate = false;
            }
        }
        If(isTrueDate)
        {
            System.out.println("The Date is Valid");
        }
        If(!isTrueDate)
        {
            System.out.println("The Date is Invalid");
        }
    }
}

```

OUTPUT

```

Microsoft Windows [Version 10.0.22000.3702]
(C) Microsoft Corporation. All rights reserved.

C:\Users\Aman\Downloads\Java Lab 2023>java Prg2
Enter Day:
31
Enter Month:
02
Enter Year:
2001
31/02/2001. The Date is Valid
C:\Users\Aman\Downloads\Java Lab 2023>

```

3. Write a JAVA program to display the following pattern.

```

      1
     2 2
    3 3 3

```

4 4 4 4
5 5 5 5 5

```
class Prg3
{
    public static void main (String args[])
    {
        System.out.println("Pattern Display");
        for (int i=1;i<=5;i++)      //Five lines
        {
            for (int j=6-i; j>1; j--) //Pre-fixed spaces 4, 3, 2, 1
            {
                System.out.print(" ");
            }

            for (int j=1;j<=i;j++)   //Printing space and number
            {
                System.out.print(" ");
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

OUTPUT

The screenshot shows a Windows Command Prompt window with the following text:

```
Microsoft Windows [Version 10.0.2600.17763]
© Microsoft Corporation. All rights reserved.

C:\Users\krishna\Documents\Java\Lab\101>javac Prg3.java
C:\Users\krishna\Documents\Java\Lab\101>java Prg3
4 4 4 4
5 5 5 5 5
```

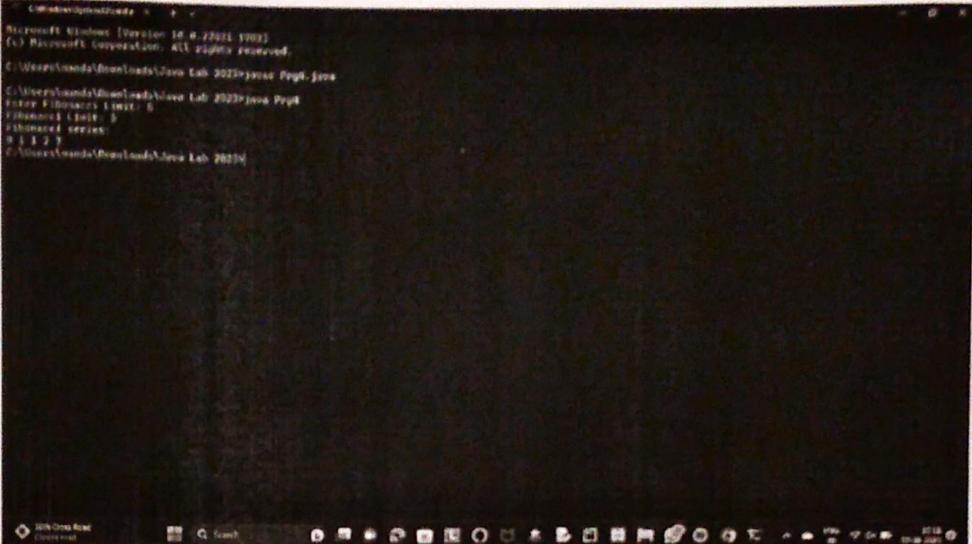
4. Write a JAVA program to print the first n members of Fibonacci series.

```
import java.util.Scanner;

public class Prg4
{
    public static void main(String args[])
    {
        int fib1=0,fib2=1,temp=0,num=0;
        Scanner sc=new Scanner(System.in); //invoke Constructor of Scanner
        System.out.print("Enter Fibonacci Limit:");
        try
        {
            num=sc.nextInt();
            if( num<=0)
            {
                throw new myexception ("Input Limit not in Range...Enter valid Limit ");
            }
            else
                System.out.println("Fibonacci Limit: "+num);
        }
        catch(Exception e)
        {
            System.out.println("Input Error—InValid Limit—Re-Enter.. "+e);
            System.exit(0);
        }
        System.out.println("Fibonacci series:");
        System.out.print(fib1);
        System.out.print(" " + fib2);
        for(int i=2;i<num;i++)
        {
            temp=fib1+fib2;
            System.out.print(" " + temp);
            fib1=fib2;
            fib2=temp;
        }
    }
}
```

```
    }  
}  
}
```

OUTPUT



```
Microsoft Windows [Version 10.0.22000.1703]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\Aman\Downloads\Java Lab 2023>java Prg5  
C:\Users\Aman\Downloads\Java Lab 2023>  
Enter first number: 5  
Enter second number: 9  
Multiplication series:  
5 10 15 20 25 30 35 40 45
```

5. Write a program to generate the multiplication tables of a range of numbers between m and n inclusive and $m < n$.

```
import java.util.Scanner;  
  
class Prg5  
  
{  
    public static void main(String args[])  
    {  
        int m=0, n=0, i, j;  
  
        System.out.println("Enter range of numbers to print their multiplication  
tables");  
        Scanner in = new Scanner(System.in);  
        try  
        {  
            m = in.nextInt();  
            n = in.nextInt();  
        }
```

```

        if( m > n || m<=0 || n<=0)
        {
            throw new Exception ("Input Numbers are not Valid...Enter valid
Numbers");
        }
        else
            System.out.print("Given Numbers: "+m+", "+n+"\n");
    }
    catch(Exception e)
    {
        System.out.println("Input Error—InValid Range—Re-Enter.. "+e);
        System.exit(0);
    }

    for (i = m; i <= n; i++)
    {
        System.out.println("\n"+ "Multiplication table of "+i);

        for (j = 1; j <= 10; j++)
        {
            System.out.println(i+"*" +j+ " = "+(i*j));
        }
    }
}

```

OUTPUT

```

Microsoft Windows [Version 10.0.22000.17507]
Copyright © Microsoft Corporation. All rights reserved.

C:\Users\Aman\Downloads\Java Lab 2020\Java Prog\

C:\Users\Aman\Downloads\Java Lab 2020\Java Prog>
java MultTab 5 6
Multiplication table of 5
5*1 = 5
5*2 = 10
5*3 = 15
5*4 = 20
5*5 = 25
5*6 = 30
5*7 = 35
5*8 = 40
5*9 = 45
5*10 = 50
Multiplication table of 6
6*1 = 6
6*2 = 12
6*3 = 18
6*4 = 24
6*5 = 30
6*6 = 36
6*7 = 42
6*8 = 48
6*9 = 54
6*10 = 60

```

6. Write a JAVA program to define a class, define instance methods for setting and retrieving values of instance variables and instantiate its object.

```
class emp
{
    int id;
    String name;
    String address;

    void setdata(int i,String n,String ad)
    {
        id=i;
        name=n;
        address=ad;
    }
    void printdata()
    {
        System.out.println("Employee details are :");
        System.out.println("ID :" +id);
        System.out.println("Name :" +name);
        System.out.println("Address :" +address);
    }
}

class Prg6
{
    public static void main(String arg[])
    {
        emp e=new emp();      //instantiate the object.
        e.setdata(123, "Shreya ", "Bangalore");
        e.printdata();
    }
}
```

OUTPUT

A screenshot of a Windows command prompt window titled "Windows PowerShell". The window shows the following text:

```
Microsoft Windows [Version 10.0.22621.3707]
Copyright © Microsoft Corporation. All Rights Reserved.

C:\Users\lalit\Downloads\Java\Lab-2023\Java_Prgs>javac Prg6.java
C:\Users\lalit\Downloads\Java\Lab-2023\Java_Prgs>java Prg6
Employee details are:
ID: 123
Name: Shreya
Address: Bangalore

C:\Users\lalit\Downloads\Java\Lab-2023\Java_Prgs>
```

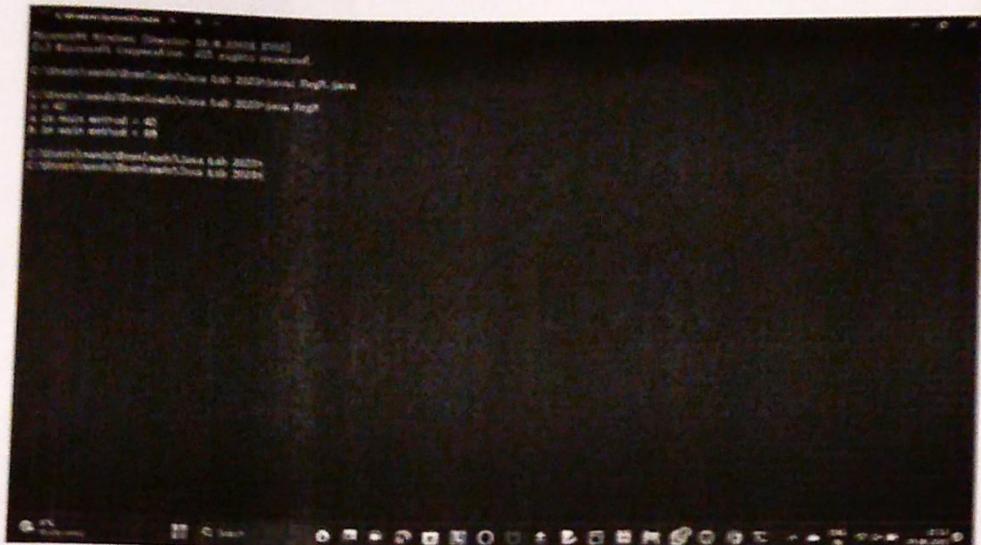
7. Write a JAVA program to demonstrate static member data and static member methods.

```
class StaticDemo
{
    static int a = 42;
    static int b = 99;
    static void callme()
    {
        System.out.println("a = " + a);
    }
}

class Prg7
{
    public static void main(String args[])
    {
        StaticDemo.callme();
        System.out.println("a in main method = " + StaticDemo.a);
    }
}
```

```
        System.out.println("b in main method = " + StaticDemo.b);
    }
}
```

OUTPUT



```
C:\Windows\system32>javac StaticDemo.java
C:\Windows\system32>java StaticDemo
b = 40
b = 40
b = 40
b = 40
```

8. Write a JAVA Program to demonstrate nested classes.

```
class outer
{
    int x = 100;
    void test()
    {
        inner i = new inner();
        i.display();
    }
    class inner           // this is an inner class
    {
        int y = 10;         // y is local to Inner
```

```
void display()
{
    System.out.println("x = " + x);
    System.out.println("y = " + y);
}
}

void showy()
{
    System.out.println("In showy");
}

class Prg8
{
    public static void main(String arg[])
    {
        outer o = new outer();
        System.out.println("From Outer");
        o.test();
        o.showy();
        outer.inner i = o.new inner();
        System.out.println("From Inner");
        i.display();
    }
}
```

OUTPUT

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\manda\Downloads\Java Lab 2023>java Prg8.java
From Outer
x = 100
From Box
x = 100
From Super
x = 100
x = 100

C:\Users\manda\Downloads\Java Lab 2023>
```

9. Write a JAVA program to demonstrate dynamic method dispatch.

```
class Box
{
    double dim1;
    double dim2;
    Box(double a, double b)
    {
        dim1 = a;
        dim2 = b;
    }
    Box(double a)
    {
        dim1 = dim2 = a;
    }
    double area()
    {
        System.out.println("Area for Box is undefined.");
        return 0;
    }
}
```

```
    }
}

class Rectangle extends Box
{
    Rectangle(double a, double b)
    {
        super(a, b);
    }

    double area()
    {
        System.out.println("Inside Area for Rectangle.");
        return dim1 * dim2;
    }
}

class Triangle extends Box
{
    Triangle(double a, double b)
    {
        super(a, b);
    }

    double area()
    {
        System.out.println("Inside Area for Triangle.");
        return dim1 * dim2 / 2;
    }
}

class Square extends Box
{
    Square (double a)
```

```
{  
    super(a);  
}  
double area()  
{  
    System.out.println("Inside Area for Square.");  
    return dim1 * dim2;  
}  
}  
class Prg9  
{  
    public static void main(String args[])  
    {  
        Box ob = new Box(10, 10);  
        Rectangle r = new Rectangle(9, 5);  
        Triangle t = new Triangle(10, 8);  
        Square s = new Square(3);  
        Box b;  
        b = r;  
        System.out.println("Area is " + b.area());  
        b = t;  
        System.out.println("Area is " + b.area());  
        b = s;  
        System.out.println("Area is " + b.area());  
        b = ob;  
        System.out.println("Area is " + b.area());  
    }  
}
```

OUTPUT

A screenshot of a Windows command prompt window titled "Windows Terminal". The command "java -version" is run, showing Java version 10.0.2. The current directory is "C:\Users\Aman\Downloads\Java Lab 2023". A Java file "Prog9.java" is compiled with the command "javac Prog9.java", which fails due to an error: "error: class names 'Prog9.java', are only accepted if annotation processing is explicitly requested". The file "Prog9.java" contains a class definition with a main method. The command "java Prog9" is then run, outputting the areas of a rectangle, triangle, and square based on user input of 45.0 for each side.

```
C:\Users\Aman\Downloads\Java Lab 2023>java -version
java 10.0.2
Java(TM) SE Runtime Environment (build 10.0.2+13)
Java HotSpot(TM) 64-Bit Server VM (build 10.0.2+13, mixed mode)

C:\Users\Aman\Downloads\Java Lab 2023>javac Prog9.java
error: class names 'Prog9.java', are only accepted if annotation processing is explicitly requested

C:\Users\Aman\Downloads\Java Lab 2023>java Prog9
Inside Area for Rectangle.
Area is 45.0
Inside Area for Triangle.
Area is 45.0
Inside Area for Square.
Area is 45.0
Area for base is undefined.
Area is 45.0
```

10. Write a JAVA program to implement inheritance and demonstrate use of method overriding.

```
class A
{
    int i,j;
    A(int a, int b)
    {
        i = a;
        j = b;
    }
    void show()
    {
        System.out.println("Inside A.. ");
        System.out.println("i and j: " + i + " " + j);
    }
}
class B extends A
{
```

```
int k;
B(int a, int b, int c)
{
    super(a, b);
    k = c;
}
void show()
{
    System.out.println("Inside B.. ");
    System.out.println("i and j: " + i + " " + j);
    System.out.println("k: " + k);
    System.out.println("Inside B.. A's Show");
    super.show();
}
}
class Prg10
{
    public static void main(String args[])
    {
        B subOb = new B(1, 2, 3);
        subOb.show();
        A Ob = new A(5, 4);
        System.out.println("\n");
        Ob.show();
    }
}
```

OUTPUT

11. Write a JAVA program to implement the concept of importing classes from user defined package and creating packages.

In P1 Folder:

```
package P1;  
public class Student  
{  
    int regno; String name;  
    public void getdata(int r, String s)  
    {  
        regno=r;  
        name=s;  
    }  
    public void putdata()  
    {  
        System.out.println("regno = " +regno);  
        System.out.println("name = " + name);  
    }  
}
```

In Root Folder

```
import P1.Student;
class Prg11
{
    public static void main(String arg[])
    {
        Student obj1=new Student();
        Student obj2=new Student();
        System.out.println("Student Details:");
        obj1.inputdata(1,"RAM");
        obj1.printdata();
        obj2.inputdata(2,"SHAM");
        obj2.printdata();

    }
}
```

OUTPUT

The screenshot shows a Windows command prompt window titled 'Windows Terminal'. The command 'java Prg11' is run, and the output displays the student details for two students, RAM and SHAM.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22021.1702]
(C) Microsoft Corporation. All rights reserved.

C:\Users\manda\Downloads\Java Lab 2023>java Prg11
Student Details:
obj1 = RAM
age = 2
name = SHAM

C:\Users\manda\Downloads\Java Lab 2023>
```

12. Write a program to demonstrate abstract class and abstract methods

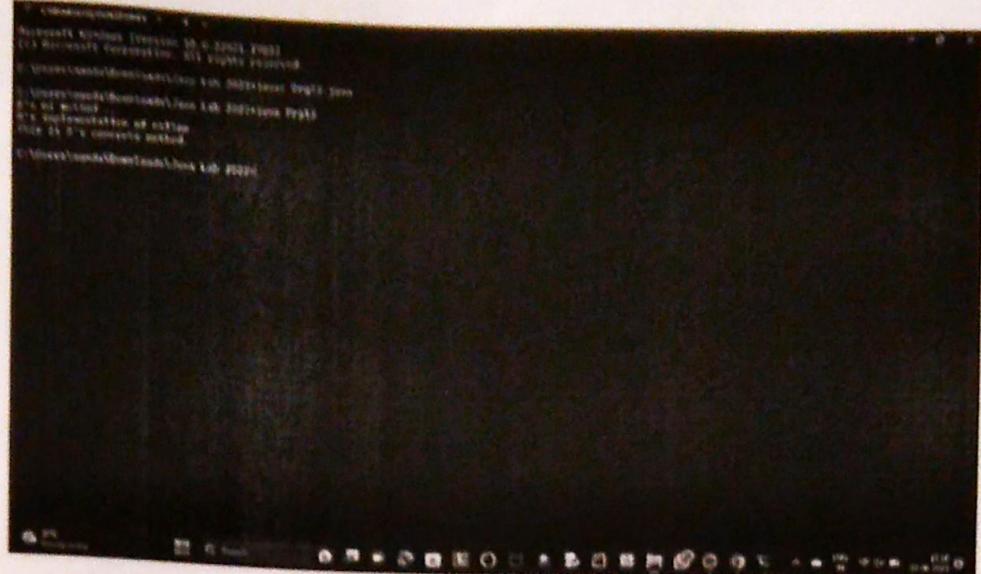
```
abstract class A
{
    abstract void callme();
    void callmetoo()      // concrete method
    {
        System.out.println("This is A's concrete method.");
    }
}

class B extends A
{
    void m1()
    {
        System.out.println("B's m1 method...");
    }

    void callme()
    {
        System.out.println("B's implementation of callme.");
    }
}

class Prg12
{
    public static void main(String args[])
    {
        B b = new B();
        b.m1();
        b.callme();
        b.callmetoo();
    }
}
```

OUTPUT



13. Write a JAVA Program to implement an array of objects of a class.

```
import java.util.Scanner;

public class Prg13
{
    public static void main(String[] args)
    {
        System.out.print("Enter Size of the Array to be created: ");
        Scanner sc = new Scanner(System.in);
        String name;
        Double sal;
        int n=sc.nextInt();

        Employee[] staff = new Employee[n];

        for (int i=0; i<n; i++)
        {
            System.out.println("Enter Array Elements(Name and Salary):");
        }
    }
}
```

```
        name=sc.next();
        sal=sc.nextDouble();
        staff[i] = new Employee(name, sal);
    }

    System.out.println("List of Employees");
    for (int j=0; j<n; j++)
        staff[j].print();
}

class Employee
{
    private String name;
    private double salary;
    public Employee(String n, double s)
    {
        name = n;
        salary = s;
    }
    public void print()
    {
        System.out.println(name + " " + salary);
    }
}
```

OUTPUT

```
Microsoft Windows [Version 10.0.22601.1702]
(C) Microsoft Corporation. All rights reserved.

C:\Users\Aman\Downloads\Java_Lab_2023\Java_Proj1.java

C:\Users\Aman\Downloads\Java_Lab_2023\Java_Proj1
Enter Size of the Array to be created: 10
Enter Array Elements( Name and Salary):
ram 10000
Enter Array Elements( Name and Salary):
vishu 20000
Enter Array Elements( Name and Salary):
salm 20000
Enter Array Elements( Name and Salary):
kiran 10000
Enter Array Elements( Name and Salary):
mangu 10000
Enter Array Elements( Name and Salary):
leela 5000
Enter Array Elements( Name and Salary):
sita 6000
Enter Array Elements( Name and Salary):
pinki 5000
Enter Array Elements( Name and Salary):
suresh 87000

List of Employees:
ram 10000.0
vishu 20000.0
salm 20000.0
sita 6000.0
kiran 10000.0
mangu 10000.0
leela 5000.0
pinki 5000.0
suresh 87000.0
```

14. Write a JAVA program to demonstrate String class and its methods.

```
import java.lang.String;

class Prg14
{
    public static void main(String arg[])
    {
        String s1=new String("Java");
        String s2="Programming language";

        System.out.println(" The string s1 is : " +s1);
        System.out.println(" The string s2 is : " +s2);

        System.out.println(" Length of the string s1 is : " +s1.length());

        System.out.println(" The first occurrence of r in s2 is at the position: "+s2.indexOf('r'));

        System.out.println(" The String in Upper Case : "
+s1.toUpperCase());
        System.out.println(" The String in Lower Case : "
+s1.toLowerCase());

        System.out.println(" s1 equals to s2 : " +s1.equals(s2));
        System.out.println(" s1 equals ignore case to s2 : " +s1.equalsIgnoreCase(s2));
    }
}
```

```
int result=s1.compareTo(s2);

System.out.println(" After compareTo()");
if(result==0) System.out.println( s1 + " is equal to "+s2);
else
if(result>0) System.out.println( s1 + " is greater than "+s2);
else
System.out.println( " " +s1 + " is smaller than " +s2);

System.out.println(" Character at an index of 3 is :"
+s1.charAt(3));

String s3=s1.substring(1,3);

System.out.println(" Extracted substring is :" +s3);

System.out.println(" After Replacing J with K in s1 : " +s1.replace('J','K'));

String s4=" This is a book      ";
System.out.println(" The string s4 is :" +s4);

System.out.println(" After trim() :" +s4.trim());
}

}
```

OUTPUT

A screenshot of a Java code editor window titled "Untitled-1". The code implements exception handling by creating a user-defined exception class "myexception" and catching it in the main method. The code is as follows:

```
class myexception extends Exception
{
    myexception(String s)
    {
        super(s);
    }
}

class Prg15
{
    public static void main(String argv[])
    {
        System.out.print("Enter marks: ");

        try
        {
            Scanner sc = new Scanner(System.in);
            int marks=sc.nextInt();
        }
        catch(myexception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

15. Write a JAVA program to implement the concept of exception handling by creating user defined exceptions.

```
import java.util.*;

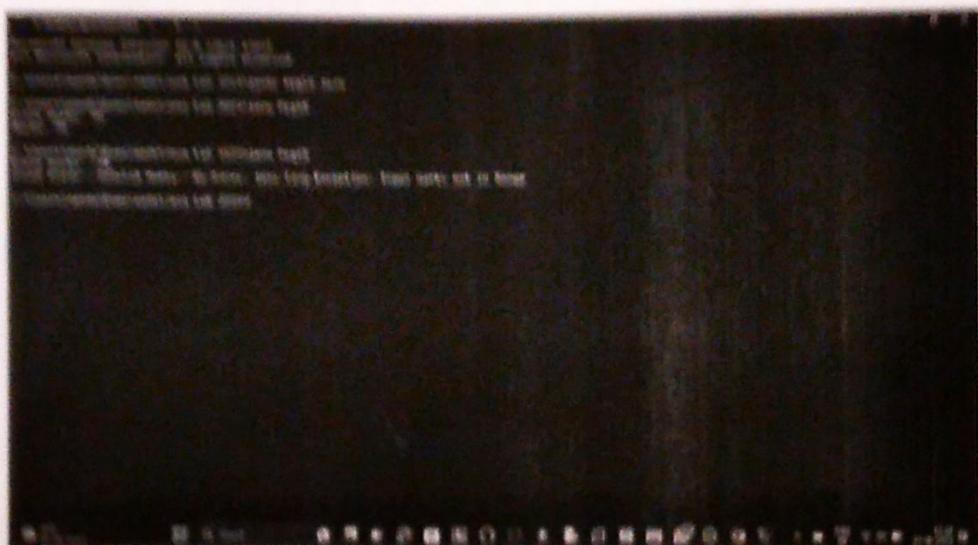
class myexception extends Exception
{
    myexception(String s)
    {
        super(s);
    }
}

class Prg15
{
    public static void main(String argv[])
    {
        System.out.print("Enter marks: ");

        try
        {
            Scanner sc = new Scanner(System.in);
            int marks=sc.nextInt();
        }
        catch(myexception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

```
if( marks > 100 || marks < 0)
{
    throw new myexception ("Input marks not in Range ");
}
else
{
    System.out.println("Marks: "+marks);
}
catch (Exception e)
{
    System.out.println("Input Error--InValid Marks--Re Enter.."+e);
}
```

OUTPUT



16. Write a JAVA program using synchronized threads, which demonstrates producer consumer concept.

```
class A  
{  
    int n;  
    boolean b=false;
```

```
synchronized int get()
{
    if(!b)
    try
    {
        wait();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    System.out.println("Got:"+n);
    b=false;
    notify();
    return n;
}

synchronized void put(int n)
{
    if(b)
    try
    {
        wait();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    this.n=n;
    b=true;
    System.out.println("Put:"+n);
    notify();
}
}

class producer implements Runnable
{
    A a1;
    Thread t1;
    producer(A a1)
    {
        this.a1=a1;
        t1=new Thread(this);
        t1.start();
    }
}
```

```
    }
    public void run()
    {
        for(int i=1;i<=10;i++)
        {
            a1.put(i);
        }
    }
}
class consumer implements Runnable
{
    A a1;
    Thread t1;
    consumer(A a1)
    {
        this.a1=a1;
        t1=new Thread(this);
        t1.start();
    }
    public void run()
    {
        for(int j=1;j<=10;j++)
        {
            a1.get();
        }
    }
}
class Prg16
{
    public static void main(String args[])
    {
        A a1=new A();
        producer p1=new producer(a1);
        consumer c1=new consumer(a1);
    }
}
```

OUTPUT

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mandal\Downloads\Java Lab 2023>java Prg16
Put:1
Get:1
Put:2
Put:3
Get:3
Put:4
Get:4
Put:5
Get:5
Put:6
Get:6
Put:7
Get:7
Put:8
Get:8
Put:9
Get:9
Put:10
Get:10

C:\Users\mandal\Downloads\Java Lab 2023>
```

17. Write a JAVA program that creates three threads. First thread displays "Good Morning" every one second, second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.

```
class goodmorning extends Thread
{
    public void run()
    {
        try
        {
            int i=0;
            while(i<5)
            {
                sleep(1000);
                System.out.println("good morning");
                i++;
            }
        }
        catch(Exception e)
        {
        }
    }
}
```

```
class hello extends Thread
{
    public void run()
    {
        try
        {
            int i=0;
            while(i<5)
            {
                sleep(2000);
                System.out.println("hello");
                i++;
            }
        }
        catch(Exception e)
        {
        }
    }
}

class welcome extends Thread
{
    public void run()
    {
        try
        {
            int i=0;
            while(i<5)
            {
                sleep(3000);
                System.out.println("welcome");
                i++;
            }
        }
        catch(Exception e)
        {
        }
    }
}
```

OUTPUT

18. Write a JAVA program which uses FileInputStream / FileOutputStream Classes.

Create new notepad file save as (t1.txt) , open t1 file write some data, save. Again, create new notepad file save as(t2.txt), don't add any content, save

```
import java.io.*;

class Prg18
{
    public static void main(String args[]) throws IOException
    {
        int i;
        FileInputStream fin;
        FileOutputStream fout;
        try
        { // open input file
            try
            {
                fin = new FileInputStream(args[0]);
            }
            catch(FileNotFoundException e)
            {
                System.out.println("Input File Not Found");
            }
        }
    }
}
```

```
    return;
}
// open output file
fout = new FileOutputStream(args[1]);
}
catch(ArrayIndexOutOfBoundsException e)
{
    System.out.println("filenames are not passed ");
    return;
}
// Copy File
try
{
    do
    {
        i = fin.read();
        if(i != -1)
            fout.write(i);
    }
    while(i != -1);
    System.out.println("File copied Successfully");
}
catch(IOException e)
{
    System.out.println("File Error");
}
fin.close();
fout.close();
}
}
```

OUTPUT

```
E:\Java prg\Java12023>java Prg18  
filenames are not passed  
  
E:\Java prg\Java12023>java Prg18 t1.txt t2.txt  
File copied Successfully  
  
E:\Java prg\Java12023>type t1.txt  
Hello Students!...  
E:\Java prg\Java12023>type t2.txt  
Hello Students!...  
E:\Java prg\Java12023>java Prg18 t3.txt t4.txt  
Input File Not Found  
  
E:\Java prg\Java12023>
```

19. Write a JAVA program to list all the files in a directory including the files present in all its subdirectories.

```
import java.io.File;  
  
public class Prg19  
{  
    static void Print(File[] arr, int index, int level)  
    {  
        //System.out.println("Index " + index + "level " + level);  
        if(index == arr.length) // exit  
            return;  
        for (int i = 0; i < level; i++)  
            System.out.print("\t");  
  
        if(arr[index].isFile())  
            System.out.println(arr[index].getName());  
  
        else if(arr[index].isDirectory())  
    }
```

```
//System.out.println("Index " + index + "level " + level);
System.out.println("[ " + arr[index].getName() + "]");
Print(arr[index].listFiles(), 0, level + 1); //for sub-directories
}

Print(arr, ++index, level);           // for main directory
}

public static void main(String[] args)
{

String dirpath = "D:/java prg/pack"; // or C://javaprof

File mdir = new File(dirpath);      // File object

if(mdir.exists() && mdir.isDirectory())
{
    File arr[] = mdir.listFiles();
    System.out.println(" The main directory Files are : " + mdir);
    Print(arr,0,0);
}
else
    System.out.println("Invalid Directory!...");
}

}
```

OUTPUT

```
W:\JAVA\J2EE\PROJECTS>javac Prg20.java
W:\JAVA\J2EE\PROJECTS>java Prg20
W:\JAVA\J2EE\PROJECTS>
```

20. Write a JAVA program to demonstrate the life cycle of applet.

```
import java.awt.*;
import java.applet.*;
// must add this code
/* <applet code="Prg20" width=500 height=300> </applet>
 */
public class Prg20 extends Applet
```

```
{  
    String msg;  
    public void init() // Called first.  
    {  
        System.out.println("Applet is initialized");// initialization  
        setBackground(Color.cyan);  
        setForeground(Color.red);  
        msg = "Applet is initialized -";  
    }  
    // Called second, after init(). Also called whenever the applet is  
    restarted.  
    public void start()  
    {  
        System.out.println("Applet is started");  
        msg = msg + " Applet is started -";  
    }  
    // Called when the applet is stopped.  
    public void stop()  
    {  
        System.out.println("Applet execution is stopped");  
        msg = msg + " Applet execution is stopped -";  
    }  
    // Called when applet is terminated. This is the last method executed.  
    public void destroy()  
    {  
        System.out.println("Applet is Destroyed"); // perform shutdown activities  
    }  
    // Called when an applet's window must be restored.
```

```
public void paint(Graphics g)
{
    System.out.println("Painting the Applet"); // redisplay contents of
    window

    msg = msg + " Painting the Applet -";
    g.drawString(msg, 20, 30);
}

}
```

OUTPUT

The screenshot displays two windows. The top window is a Command Prompt titled 'Command Prompt - Appletviewer prg20.java'. It shows the following terminal session:

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

C:\Users\Anbika>cd\
C:\>e:
E:\>cd java_prg
E:\java_prg>javac prg20.java
E:\java_prg>appletviewer prg20.java
Applet is initialized
Applet is started
Painting the Applet
Painting the Applet
```

The bottom window is titled 'Applet Viewer: prg20' and contains the text:

Applet
Applet is initialized -- Applet is started -- Painting the Applet --
Applet started.