

Note :I am not responsible for any mistakes in this document

**WRITE ALL AT RIGHT
SIDE ONLY**

-SETHU

INDEX

Name of the Experiment: Factorial Number

1. Write a program in Java to execute factorial of given number.

AIM:

TO perform the execution of a program that calculates the factorial of a given number using Java.

Source code:

```

import java.util.Scanner;
class factorial
{
    public static void main (String args[])
    {
        int n, fact=1, a=1;
        Scanner s = new Scanner (System.in);
        System.out.println ("enter n value:");
        n = s.nextInt();
        while (a <= n)
        {
            fact = fact * a;
            a++;
        }
        System.out.println ("the factorial of given number = " + fact);
    }
}

```

Name of the Experiment :

Page No.: 2

Date:

Practical No.:

OUTPUT:

enter n value : 6

the factorial of given number = 720 .

Name of the Experiment: Method overloading.

2. Write a program in java using method overloading.

AIM:

to perform the execution of a program that calculate area & parameter of rectangle in java.

SOURCE CODE:-

```

class rectangle
{
    float l,b;
    void rect(float l, float b)
    {
        this.l=l;
        this.b=b;
        float ar=l*b;
        float p=2*(l+b);
        System.out.println ("area of rectangle = "+ar);
        System.out.println ("perimeter = "+p);
    }
    void rect(int l, int b)
    {
        this.l=l;
        this.b=b;
        float ar=l*b;
        float p=2*(l+b);
        System.out.println ("area of rectangle = "+ar);
        System.out.println ("perimeter = "+p);
    }
}

```

Name of the Experiment :

```
void rect (float l)
{
    this.l=l;
    b=1;
    System.out.println ("area of rectangle = "+(l*b));
    System.out.println ("perimeter = "+2*(l+b));
}

public static void main (String args [])
{
    rectangle v=new rectangle ();
    v.rect (4.2F);
    v.rect (4,5);
    v.rect (4.2F, 2.5F);
}
```

Name of the Experiment :

Page No.: 5

Date:

Practical No.:

OUTPUT :-

Area of rectangle = 17.639997

Perimeter = 16.8

Area of rectangle = 20.0

Perimeter = 18.0

Area of rectangle = 10.5

Perimeter = 13.04.

Name of the Experiment: Constructor

3. Write a program in java using constructor.

AIM :-

To perform the execution of a program that calculate area & Perimeter of rectangle in java. using constructor.

SOURCE CODE :-

```

class rectangle
{
    float l,b;
    rectangle (float l, float b)
    {
        this.l=l;
        this.b=b;
        float ar=l*b;
        float p=2*(l+b);
        System.out.println ("area of rectangle = "+ar);
        System.out.println ("Perimeter = "+p);
    }
    rectangle (int l, int b)
    {
        this.l=l;
        this.b=b;
        float ar=l*b;
        float p=2*(l+b);
        System.out.println ("area of rectangle = "+ar);
    }
}

```

Name of the Experiment :

```
System.out.println ("perimeter = "+p);
```

```
{  
    rectangleC (float l)  
}
```

```
{  
    this.l=l;
```

```
b=l;
```

```
System.out.println ("area of rectangle = "+(l*b));
```

```
System.out.println ("perimeter = "+2*(l+b)));
```

```
{
```

```
public static void main (String args [])
```

```
{  
    rectangleC v1 = new rectangleC (4, 5);
```

```
    rectangleC v2 = new rectangleC (4.2F);
```

```
    rectangleC v3 = new rectangleC (4.2F, 5.2F);
```

```
{
```

```
}
```

Name of the Experiment :

OUTPUT :-

$$\text{area of rectangle} = 20.0$$

$$\text{Perimeter} = 18.0$$

$$\text{area of rectangle} = 17.639997$$

$$\text{Perimeter} = 16.8$$

$$\text{area of rectangle} = 21.839997$$

$$\text{Perimeter} = 18.8$$

Name of the Experiment: Method overriding.

A. Write a program in java using Method overriding.

AIM :-

To perform the execution of an addition program using method overriding.

SOURCE CODE :-

```

class mysuper
{
    int a,b;
    mysuper (int a , int b)
    {
        this.a = a;
        this.b = b;
    }
    void show()
    {
        System.out.println ("sum of two numbers = "+(a+b));
    }
}

```

```

class mysub1 extends mysuper
{

```

```

    int c;
    mysub1 (int a, int b , int c)
    {
        super(a,b);
        this.c = c;
    }
    void show()

```

```
{  
System.out.println ("sum of three numbers = "+(a+b+c));  
}  
  
{  
class mysub2 extends mysuper  
{  
int d;  
mysub2 (int a, int b, int c, int d)  
{  
super (a,b);  
this.c=c;  
this.d=d;  
}  
void show()  
{  
System.out.println ("sum of four numbers = "+(a+b+c+d));  
}  
public static void main (String [] args)  
{  
mysuper obj1 = new mysuper (5,5);  
mysuper obj2 = new mysub1 (5,5,5);  
mysuper obj3 = new mysub2 (5,5,5,5);  
obj1.show();  
obj2.show();  
obj3.show();  
}
```

Name of the Experiment :

Page No.: 7

Date:

Practical No.:

?
?.

OUTPUT :-

Sum of two number = 10.

sum of three numbers = 15

sum of four number = 20.

Name of the Experiment: Multiple catch.

Q. write a program in java using multiple catch.

AIM:-

To perform the execution of a program using multiple catch.

SOURCE CODE :-

class myexcept

{

public static void main (String args [])

{

int a= Integer.parseInt (args [0]);

int b= Integer.parseInt (args [1]);

int c [] = { 30, 40, 25 };

try

{

int r=a/b;

System.out.println ("the result = "+r);

System.out.println ("An array value = "+c [2]);

}

catch (ArithmaticException e)

{

System.out.println (" can not divisible by zero...");

}

catch (ArrayIndexOutOfBoundsException ae)

{

System.out.println ("An Array Index out of bound ...");

Name of the Experiment :

Page No.: 13

Date:

Practical No.:

```
}

finally
{
    System.out.println ("the program is completed");
}

}
```

Compile process :-

javac myexcept.java

java myexcept 10 5 (must give two arguments)

OUTPUT:-

The result = 2

An array value = 25

The program is completed

6. write a program in java to compute arithmetic calculation using packages.

AIM:-

To compute arithmetic calculation using packages.

SOURCE CODE :-

Notebook 1

```
package arithmetic;
public class addition
{
    int a,b;
    public void compute (int a, int b)
    {
        this.a=a;
        this.b=b;
        System.out.println ("addition of two values "+(a+b));
    }
}
```

save: addition.java

compile: javac -d. addition.java

Notebook 2

```
package arithmetic;
public class subtraction
{
    int a,b;
    public void compute (int a, int b)
    {
    }
```

```
this.a=a;  
this.b=b;  
System.out.println ("Subtraction of two values = "+(a-b));  
}
```

Save : subtraction.java.

Compile : javac -d . subtraction.java.

Notebook 3

```
package arithmetic;  
public class multiplication
```

{

```
int a,b;
```

```
public void compute (int a, int b)
```

{

```
this.a=a;
```

```
this.b=b;
```

```
System.out.println ("Multiplication of two values = "+(a*b));
```

}

}

Save : multiplication.java.

Compile : javac -d . multiplication.java.

Notebook 4.

```
package arithmetic;  
public class division
```

{

```
int a,b;  
public void compute (int a, int b)  
{  
    this.a=a;  
    this.b=b;  
    System.out.println (" division of two values = " + (a/b));  
}
```

Save : division.java.

Compile : java c -d division.java.

Note pad 5

```
import arithmetic.addition;  
import arithmetic.subtraction;  
import arithmetic.multiplication;  
import arithmetic.division;  
class ex  
{  
    public static void main (String args [ ] )  
    {  
        addition obj1 = new addition ();  
        obj1.compute (10,5);  
        subtraction obj2 = new subtraction ();  
        obj2.compute (10,5);  
        multiplication obj3 = new multiplication ();  
        obj3.compute (10,5);  
        division obj4 = new division ();
```

Name of the Experiment:

Page No.: 17

Date:

Practical No.:

obj4. compute(10,15);

}

}

Save : ex.java.

compile : javac ex.java.

java ex.

OUTPUT :-

addition of two values = 15

subtraction of two values = 5

multiplication of two values = 50

division of two values = 2.

Name of the Experiment: Binary search.

Q. Write a program in Java using binary search.

AIM:- To perform binary search.

SOURCE CODE :-

```

import java.util.Scanner;
class Binary Search
{
    public static void main (String args[])
    {
        int a [] = new int [50];
        int i, search, n, first, middle, last;
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter n value : ");
        n = s.nextInt ();
        System.out.println ("Enter the elements in sorted order : ");
        for (i=0; i<n; i++)
        {
            a [i] = s.nextInt ();
        }
        System.out.println ("Enter the Element to search : ");
        search = s.nextInt ();
        first = 0;
        last = n-1;
    }
}

```

while (first <= last)

$$\{ \text{middle} = (\text{first} + \text{last}) / 2;$$

$\text{if } (\text{a}[\text{middle}] < \text{search})$

{ first = middle + 1 ;

else if ($a[middle] == search$)

System.out.println("Search + " is found in the list at position " + (middle+1));

break;

3

last = middle - 1;

۲

3

if (first > last)

5

```
System.out.println("Search + " is not found in the list");
```

3

Name of the Experiment :

Page No.: 20

Date:

Practical No.:

```
s.close();  
}  
}
```

OUTPUT :-

Enter n value = 5

Enter the elements in sorted order:

9

867

5

65

3456

Enter the element to search: 3456

3456 is found in the list at position 5

Name of the Experiment: Bubble sort.

8. Write a program in java using bubble sort.

AIM:-

to perform bubble sort

SOURCE CODE :-

```

import java.util.Scanner;
class bubblesort
{
    public static void main (String args [])
    {
        int a[] = new int [50];
        int n,i;
        Scanner s = new Scanner (System.in);
        System.out.println ("In Enter the number elements for
                           sorting\n");
        n = s.nextInt ();
        System.out.println ("In Enter An Element\n");
        for (i=0; i<n; i++)
        {
            a[i] = s.nextInt ();
        }
        System.out.println ("In Before sorting\n");
        for (i=0; i<n; i++)
        {
            System.out.println ("In " + a[i]);
        }
    }
}

```

Name of the Experiment :

```

    }

    Bsort(n,a);
    System.out.println ("In after sorting\n");
    for (i=0; i<n; i++)
    {
        System.out.println ("\n"+a[i]);
    }

public static void Bsort (int n, int a[])
{
    int i,j,temp;
    for (i=0; i<n-1; i++)
    {
        for (j=0; j<n-i-1; j++)
        {
            if (a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}

```

Page No.: 23

Date:

Practical No.:

Name of the Experiment :

OUTPUT :-

Enter the number elements for sorting.

5

Enter an element

12

9

456

6453

34566

Before sorting.

12

9

456

6453

34566

After sorting.

9

12

456

6453

3456 6

Q. Write a program in Java using insertion sort.

AIM :-

To perform insertion sort.

SOURCE CODE :-

```

import java.util.Scanner;
class insertion sort
{
    public static void main (String args[])
    {
        int a[] = new int[50];
        int n;
        Scanner s = new Scanner (System.in);
        System.out.println ("In Enter the number of elements for sorting (n)");
        n = s.nextInt();
        System.out.println ("In Enter an element (n)");
        for (i=0; i<n; i++)
        {
            a[i] = s.nextInt();
        }
        System.out.println ("In Before sorting (n)");
        for (i=0; i<n; i++)
        {
            System.out.println ("In " + a[i]);
        }
    }
}

```

Name of the Experiment :

```

    : sort(n,a);
System.out.println("In After sorting\n");
for(i=0; i<n; i++)
{
    System.out.println("In "+a[i]);
}
}

public static void isort(int n, int a[])
{
    int t, i, j;
    for(i=1; i<n; i++)
    {
        t = a[i];
        j = i - 1;
        while (j >= 0 && a[j] > t)
        {
            a[j+1] = a[j];
            j--;
        }
        a[j+1] = t;
    }
}

```

Name of the Experiment :

Page No.: 26

Date:

Practical No.:

OUTPUT:-

Enter the number of elements for sorting.

5

Enter an element

9

8

7

6

5

Before sorting.

9

8

7

6

5

After sorting.

5

6

7

8

9

Q. Write a program in java using stack.

AIM:-

To perform stack program.

SOURCE CODE :-

```

import java.util.Scanner;
class stack
{
    int stk[], top, max=5;
    Scanner s = new Scanner(System.in);
    stack()
    {
        stk = new int [max];
        top=-1;
    }
    void push()
    {
        if (top == max-1)
        {
            System.out.println("stack is overflow\n");
            return;
        }
        else
        {
            top = top+1;
            System.out.println("Enter an element\n");
            stk [top] = s.nextInt();
        }
    }
}

```

Name of the Experiment :

```
System.out.println ("Insertion successfully.\n");
    return;
}
}

void pop()
{
    if (top == -1)
    {
        System.out.println ("stack is underflow\n");
        return;
    }
else
{
    System.out.println ("In Deleted item is ==> " + stk[top]);
    top = top - 1;
    System.out.println ("Deleted successfully\n");
    return;
}
}

void show()
{
    if (top == -1)
    {
        System.out.println ("In stack is underflow\n");
        return;
    }
}
```

Name of the Experiment :

```

else
{
    int temp;
    for (temp = top; temp >= 0; temp--)
    {
        System.out.println (stack[temp]);
    }
    return;
}

void menu()
{
    int choice;
    do
    {
        System.out.println ("1. **** MAIN MENU ****\n");
        System.out.println ("1. PUSH() operation\n");
        System.out.println ("2. POP() operation\n");
        System.out.println ("3. SHOW() operation\n");
        System.out.println ("4. EXIT() operation\n");
        System.out.println ("Enter your choice\n");
        choice = s.nextInt();
        switch (choice)
        {
            case 1 : push();
        }
    }
}

```

```
        break;  
case 2 : Pop();  
        break;  
case 3 : Show();  
        break;  
default : System.out.println("Sorry! wrong option\n");  
}  
} while (choice != 4);  
}  
public static void main (String args [])  
{  
    Stack obj = new Stack ();  
    obj.menu();  
}
```

OUTPUT :-

***** MAIN MENU *****

1. PUSH() operation.
2. POP() operation
3. SHOW() operation.
4. EXIT() operation.

Enter your choice.

1
Enter an element

Name of the Experiment :

Insertion successfully

***** MAIN MENU *****

1. PUSH () operation.
2. POP () operation
3. SHOW () operation
4. EXIT () operation.

Enter your choice

3

369

***** MAIN MENU *****

1. PUSH () operation
2. POP () operation
3. SHOW () operation
4. EXIT () operation

Enter your choice.

2

Deleted item is ==> 369

Deletion successfully.

***** MAIN MENU *****

1. PUSH () operation
2. POP () operation
3. SHOW () operation.
4. EXIT () operation.

Enter your choice.

Sorry! wrong option.

Name of the Experiment: Queue.

11. write a program in java using queue.

AIM :-

To perform Queue program.

SOURCECODE :-

```
import java.util.Scanner;
class queue
```

{

```
int Q[], front, rear, max = 5;
```

```
Scanner s = new Scanner (System.in);
```

queue()

{

```
Q = new int [max];
```

}

void enqueue ()

{

```
if (rear == max - 1)
```

{

```
System.out.println ("Queue is full....Can't insert any element\n");
```

return;

}

else

{

```
rear = rear + 1;
```

```
System.out.println ("Enter the element to insert : ");
```

```
Q [rear] = s.nextInt();
```

```
if (front == -1)
```

{

Name of the Experiment:

Practical No.:

```
front = 0;
}
return;
}

void dequeue()
{
    if (front == -1 && rear == -1)
    {
        System.out.println("Queue is empty...Can't delete.\n");
        return;
    }
    else
    {
        System.out.println("The deleted item => " + Q[front]);
        front = front + 1;
        System.out.println("deletion successfully\n");
        if (rear + 1 == front)
        {
            front = rear = -1;
        }
        return;
    }
}

void show()
{
    if (front == -1 && rear == -1)
    {
```

Name of the Experiment :

```
System.out.println ("Queue is empty... can't show \n");
    return;
```

{
else

```
{  
    int temp;  
    System.out.println ("Queue element are :\n");
```

```
for (temp=front; temp <= rear; temp++)
```

{

```
    System.out.println (a [temp]);
```

{

```
return;
```

{

{

```
void menu ()
```

{

```
int choice;
```

do

{

```
    System.out.println ("\n*** MAIN MENU ***\n");
```

```
    System.out.println (" 1. Enqueue operation\n");
```

```
    System.out.println (" 2. Dequeue operation\n");
```

```
    System.out.println (" 3. Show operation\n");
```

```
    System.out.println (" 4. Exit operation\n");
```

```
    System.out.println ("Enter your choice.\n");
```

Name of the Experiment :

```

choice = s.nextInt();
switch (choice) {
    case 1 : Enqueue();
        break;
    case 2 : Dequeue();
        break;
    case 3 : show();
        break;
    default : System.out.println ("sorry ! wrong option\n");
}
while (choice != 4);
}

public static void main (String args[])
{
    queue obj = new queue ();
    obj.menu();
}
}

```

OUTPUT:-

*** MAIN MENU ***

1. Enqueue operation
2. Dequeue operation
3. Show operation.
4. Exit operation.

Enter your choice

Enter the element to insert :

69

*** MAIN MENU ***

1. Enqueue operation
2. Dequeue operation
3. Show operation
4. Exit operation

Enter your choice .

3

Queue elements are :

69

*** MAIN MENU ***

1. Enqueue operation
2. Dequeue operation
3. Show operation
4. Exit operation

Enter your choice

2

The deleted item ==> 69

deletion successfully.

*** MAIN MENU ***

1. Enqueue operation.
2. Dequeue operation.
3. Show operation.
4. Exit operation.

Enter your choice .

4

Sorry ! wrong option .

12. Write a program in java to perform addition of two numbers by using applet.

AIM:-

Perform addition of two numbers by using applet

SOURCE CODE :- notepad-1

```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class app1 extends Applet implements ActionListener {
    TextField t1 = new TextField(10);
    TextField t2 = new TextField(10);
    TextField t3 = new TextField(10);
    Label l1 = new Label("Enter A value");
    Label l2 = new Label("Enter B value");
    Label l3 = new Label("Result:");
    Button b1 = new Button("Addition");

    public void init() {
        setBackground(Color.white);
        add(l1); add(t1); add(l2); add(t2); add(l3);
        add(t3); add(b1);
        b1.addActionListener(this);
    }
}

```

Name of the Experiment :

```

public void actionPerformed (ActionEvent e)
{
    if (e.getSource () == bt)
    {
        int a = Integer.parseInt (t1.getText ());
        int b = Integer.parseInt (t2.getText ());
        t3.setText (" " + (a+b));
    }
}

```

/< applet code = "appl.class" width = 500 height = 100 >
</applet>/

Save : appl.java.

notepad 2

<html>

<body>

<applet code = "appl.class" width = "500"
height = "100"></applet>

</body>

</html>

Save : appl.htm

Compile : javac appl.java.

Run : appletviewer appl.htm.

Page No.: 39

Date:

Practical No.:

Name of the Experiment :

OUTPUT :-

Enter A value

3

Enter B value

6

Result :

9

Addition