SIMPLE JAVA PROGRAM - TO DISPLAY THE MESSAGE AS WELCOME USER NAME

Program

```
import java.lang.*;
import java.util.*;
class Welcome
{
	public static void main(String args[])
	{
		Scanner obj=new Scanner(System.in);
		System.out.println("Enter your name");
		String s = obj.nextLine();
		System.out.println("WELCOME " + s);
	}
}
```

Sample Output

```
Enter your name
Ram
WELCOME Ram
```

SWAPPING OF TWO NUMBERS

Program

```
import java.util.Scanner;
public class Swapping
 {
int num1,num2, temp;
void getInput(){
   Scanner obj=new Scanner(System.in);
   System.out.println("Enter the number 1:");
   num1=obj.nextInt();
   System.out.println("Enter the number 2:");
   num2=obj.nextInt();
 }
 void display()
 {
    System.out.println(" Number 1 value is: "+num1);
    System.out.println(" Number 2 value is: "+num2);
 }
 void swap()
 {
  temp=num1;
    num1=num2;
    num2=temp;
 }
 public static void main(String args[])
 {
 Swapping s=new Swapping();
    s.getInput();
    System.out.println("BeforeSwapping");
    s.display();
    s.swap(); //to perform swapping
    System.out.println("After Swapping");
    s.display();
 }
}
```

Sample Output:

```
Enter the number 1:45
Enter the number 2:34
Before Swapping
Number 1 value is: 45
Number 2 value is: 34
After Swapping
Number 1 value is: 34
Number 2 value is: 45
```

CALCULATE PERIMETER, AREA AND VOLUME OF ANY ONE OF THE SHAPE

Program

```
import java.util.Scanner;
public class Shape
{
int side;
    void getInput()
{
    System.out.println("Enter the side of Square object:");
    Scanner obj=new Scanner(System.in);
    side=obj.nextInt();
    }
    void areaSquare()
    {
       System.out.println("The area of Square Object is:"+side*side);
    }
    void perimeterSquare()
    {
       System.out.println("The perimeter of Square Object is:"+4*side);
    }
    void cube()
    {
       System.out.println("The volume of cube is:"+side*side*side);
    }

    public static void main(String args[])
    {
    Shape e=new Shape ();
    e.getInput();
    e.areaSquare();
    e.perimeterSquare();
    e.cube();
    }
}
```

Sample Output:

```
Enter the side of Square object:
5
The area of Square Object is: 25
The perimeter of Square Object is:
20 The volume of cube is:125
```

ELIGIBILITY FOR VOTING OR NOT

Program

```
import java.util.Scanner;
public class VoterDemo
{
   int age;
   String name;
   void getVoterDetails()
   {
      Scanner obj=new Scanner(System.in);
      System.out.println("Enter the Person Name:");
      name=obj.nextLine();
      System.out.println("Enter the Person Age:");
      age=obj.nextInt();
   }
   void checkEligible()
   {
   if(age >=18 )
         System.out.println(name+" is eligible to Vote");
      else
          System.out.println(name+" is not eligible to Vote");
   }

   public static void main(String args[])
   {
    VoterDemo v=new VoterDemo();
    v.getVoterDetails();
    v.checkEligible();
   }
}
```

Sample Outputs:

```
Enter the Person Name:Ravi
Enter the Person Age: 23
Ravi is eligible to Vote

Enter the Person Name:Kavi
Enter the Person Age:17
Kavi is not eligible to Vote
```

SUM OF NATURAL NUMBERS UPTO THE LIMIT

Program

```
 import java.util.Scanner;
public class SumFirstN
{
int calculate(int n)
{
  intsum=0;
 for(int i=1;i<=n;i++)
{
          sum+=i;
         return sum;
        }
        public static void main(String arg[])
        {
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the Limit (N):");
        intn=obj.nextInt();
SumFirstN s=new SumFirstN();
System.out.println("The sum of First N Natural number is:"+ s.calculate(n));
}
}
```

Outputs:

```
Enter the Limit(N): 10
The sum of First N Natural number is:55
```

ADDITIONAL EXERCISES

1. Write a java program to find simple interest and compound interest respectively p*n*r/100,$p(1+r/100)^n$?
2. Write a java program to convert Fahrenheit to Centigrade and vice versaf=9/5*c+32
3. Write a Java program to add two numbers without using '+' symbol [Note ++, -- operator or (unary minus)]
4. Write a java program to compute the sum of this geometric progression $1+x+x^2+x^3+ \quad +x^n$.
5. Write a java program to find factorial, NCR =n!/r!(n-r)! , NPR=n!/(n-r)!
6. Write a java program to check whether the given number is odd oreven.
7. Write a java program to check whether the given year is leap year or not.
8. Write a java program to check whether the given characters is vowel orconsonant.
9. Write a java program to find smallest among threenumbers.
10. Write a java program to find largest among threenumbers.
11. Write a java program to get the student marks and print the grade respectively
Example <50 --- RA, 50 – 60 --- B, 61 – 70 --- B+, 71-80 --- A, 81-90 ---A+,91-100 O.
12. Write a java program to print the day of the week (0 – Sun, 1 – Mon, …, 6 – Sat).

Ex. No: 1	SORTING AND SEARCHING ALGORITHMS
Date:	

A) Sequential Search

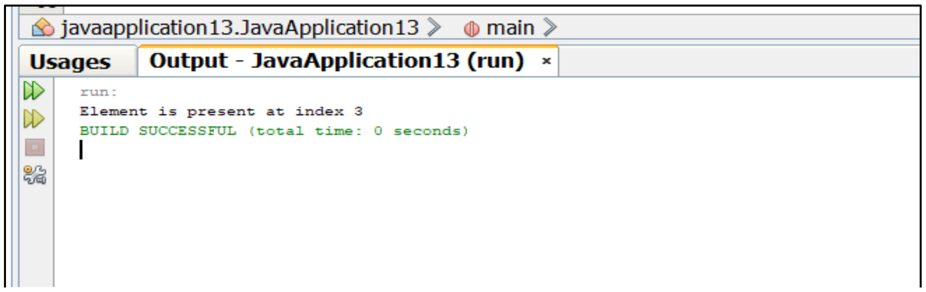
Program:

```
import java.util.*;
class sequential
  {
 // Function for linear search
 public static int search(int arr[], int x)
 {
        int n = arr.length;
        // Traverse array arr[]
        for (int i = 0; i < n; i++)
         {
        // If element found then
        // return that index
        if (arr[i] == x)
        return i;
        }
        return -1;
 }

 // Driver Code

 public static void main(String args[])
 {
        // Given arr[]
        int arr[] = { 2, 3, 4, 10, 40 };
 // Element to search
        int x = 10;
        // Function Call
        int result = search(arr, x);
        if (result == -1)
        System.out.print("Element is not present in array");
        else
        System.out.print("Element is present"+ " at index "+ result);
 }
 }
```

Output:


javaapplication13.JavaApplication13 > main >
Usages
Output - JavaApplication13 (run)

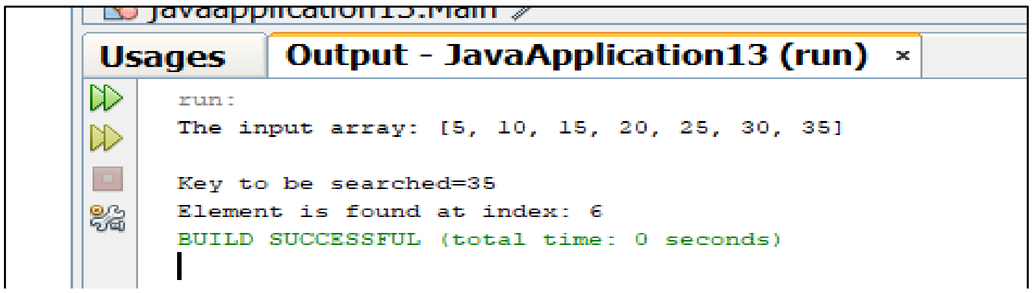
```
run:
Element is present at index 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

B) Binary Search

Program:

```
import java.util.*;
class Main
{
public static void main(String args[])
{
int numArray[] = {5,10,15,20,25,30,35};
System.out.println("The input array: " + Arrays.toString(numArray));
int key = 20;    //key to be searched
System.out.println("\n Key to be searched=" + key);
   //set first to first index
int first = 0;
   //set last to last elements in array
int last=numArray.length-1;
   //calculate mid of the array
int mid = (first + last) /2;
   //while first and last do not overlap
while( first <= last )
  {
  //if the mid < key, then key to be searched is in the first half of array
if ( numArray[mid] < key ){
first = mid + 1;
 }
else if ( numArray[mid] == key )
{
//if key = element at mid, then print the location
System.out.println("Element is found at index: " + mid);
break;
  }
else
{
 //the key is to be searched in the second half of the array
last = mid - 1;
   }
mid = (first + last)/2;
 }
//if first and last overlap, then key is not present in the array
if ( first > last )
  {
System.out.println("Element is not found!");
}
 }
}
```

Output:



```
run:
The input array: [5, 10, 15, 20, 25, 30, 35]

Key to be searched=35
Element is found at index: 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

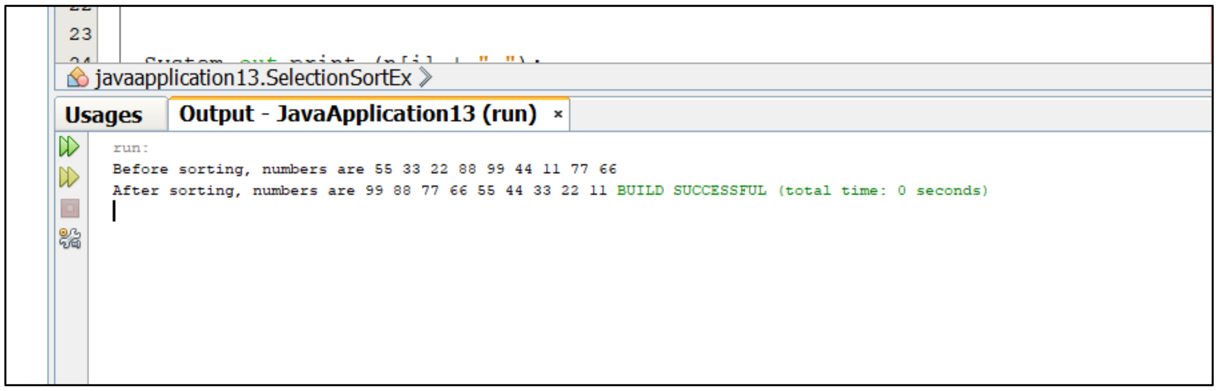
C) SELECTION SORT

Program:

```
 public class SelectionSortEx
 {
 public static void main(String a[])
  {
  //Numbers which are to be sorted
 int n[] = {55,33,22,88, 99,44,11, 77,66 };
 //Displays the numbers before sorting
 System.out.print("Before sorting, numbers are ");
 for (int i = 0; i < n.length; i++) {
 System.out.print(n[i] + " ");
   }
 System.out.println();
//Sorting in ascending order using bubble sort
 initializeselectionSort(n);
        //Displaying the numbers after sorting
 System.out.print("After sorting, numbers are ");
 for (int i = 0; i < n.length; i++) {

 System.out.print(n[i] + " ");
        }
    }
 //This method sorts the input array in descending order
 public static void initializeselectionSort(int n[])
 {
 int i, j, first, temp;
 for (i = n.length - 1; i > 0; i--) {
            first = 0; //initialize to subscript of first element
            for (j = 1; j <= i; j++) //locate smallest element between 1 and i.
            {
               if (n[j] < n[first])
 first = j;
            }
            temp = n[first]; //swap the smallest found in position i.
 n[first] = n[i];
 n[i] = temp;
        }
    }
 }
```

Output:


javaapplication13.SelectionSortEx
Usages
Output - JavaApplication13 (run)

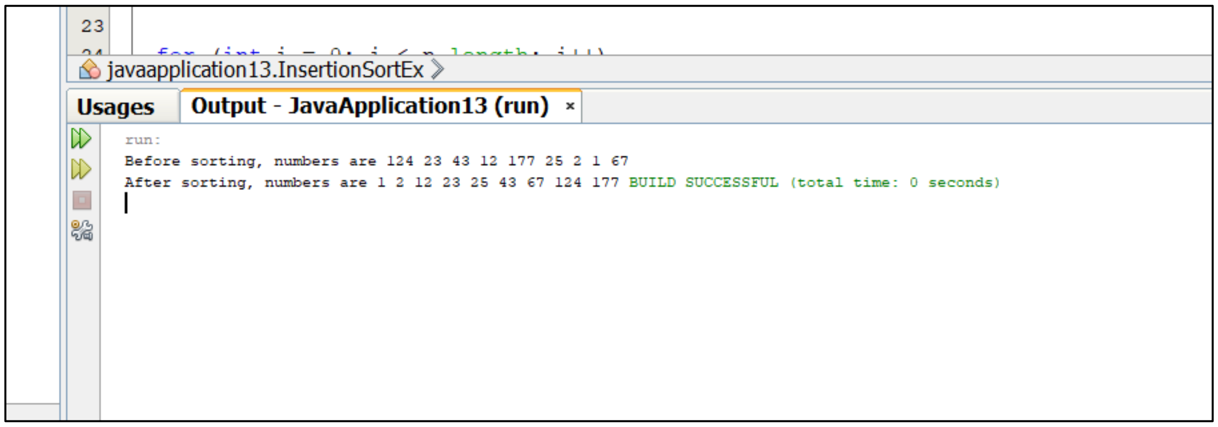
```
run:
Before sorting, numbers are 55 33 22 88 99 44 11 77 66
After sorting, numbers are 99 88 77 66 55 44 33 22 11 BUILD SUCCESSFUL (total time: 0 seconds)
```

D) INSERTION SORT

Program:

```
public class InsertionSortEx
{
public static void main(String a[])
{
        //Numbers which are to be sorted
int n[] = {124,  23,  43, 12,  177,  25, 2, 1,67};
        //Displays the numbers before sorting
System.out.print("Before sorting, numbers are ");
for (int i = 0; i < n.length; i++)
{
System.out.print(n[i] + " ");
        }
System.out.println();
        //Sorting in ascending order using bubble sort
initializeInsertionSort(n);
        //Displaying the numbers after sorting
System.out.print("After sorting, numbers are ");
for (int i = 0; i < n.length; i++)
{
System.out.print(n[i] + " ");
        }
}
 //This method sorts the input array in asecnding order
public static void initializeInsertionSort(int n[])
{
for (int i = 1; i < n.length; i++)
 {
int j = i;
int B = n[i];
while ((j > 0) && (n[j - 1] > B))
 {
n[j] = n[j - 1];
j--;
  }
n[j] = B;
        }
    }
}
```

Output:


javaapplication13.InsertionSortEx
Usages
Output - JavaApplication13 (run)

```
run:
Before sorting, numbers are 124 23 43 12 177 25 2 1 67
After sorting, numbers are 1 2 12 23 25 43 67 124 177 BUILD SUCCESSFUL (total time: 0 seconds)
```

Expt. No: 2	
Date:	STACK & QUEUE IMPLEMENTATION

A) STACK IMPLEMENTATION

Program:

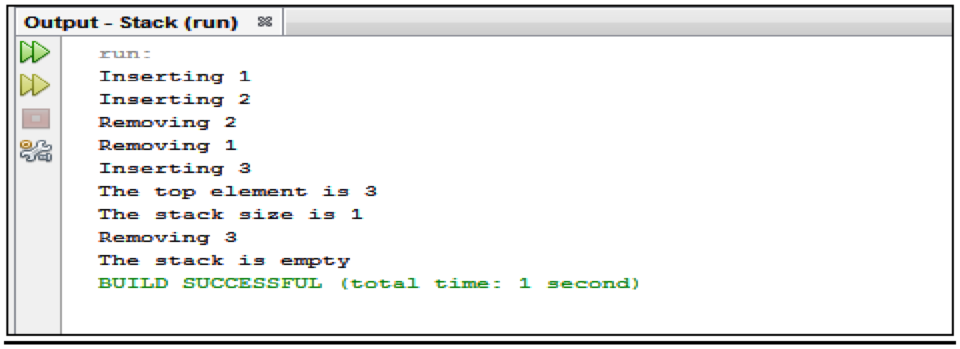
```
class Stack
   {
   private int arr[];
   private int top;
   private int capacity;
        // Constructor to initialize the stack
   Stack(int size)
       {
   arr = new int[size];
   capacity = size;
   top = -1;
       }
       // Utility function to add an element `x` to the stack
   public void push(int x)
       {
   if (isFull())
           {
   System.out.println("Overflow\nProgram Terminated\n");
   System.exit(-1);
           }
   System.out.println("Inserting " + x);
   arr[++top] = x;
       }
        // Utility function to pop a top element from the stack
   public int pop()
       {
          // check for stack underflow
   if (isEmpty())
       {
   System.out.println("Underflow\nProgram Terminated");
   System.exit(-1);
           }
   System.out.println("Removing " + peek());
            // decrease stack size by 1 and (optionally) return the popped element
   return arr[top--];
       }
```

```
    // Utility function to return the top element of the stack
public int peek()
    {
if (!isEmpty())
{
return arr[top];
        }
        else
{
System.exit(-1);
        }
return -1;
    }
    // Utility function to return the size of the stack

public int size()
 {
return top + 1;
    }
    // Utility function to check if the stack is empty or not
public boolean isEmpty()
{
return top == -1;              // or return size() == 0;
    }
    // Utility function to check if the stack is full or not
public boolean isFull()
 {
return top == capacity - 1;     // or return size() == capacity;
    }
}
class Main
{
public static void main (String[] args)
    {
Stack stack = new Stack(3);
stack.push(1);      // inserting 1 in the stack
stack.push(2);      // inserting 2 in the stack
stack.pop();        // removing the top element (2)
stack.pop();        // removing the top element (1)
stack.push(3);      // inserting 3 in the stack
System.out.println("The top element is " + stack.peek());
System.out.println("The stack size is " + stack.size());
stack.pop();        // removing the top element (3)
```

```
  // check if the stack is empty
 if (stack.isEmpty()) {
 System.out.println("The stack is empty");
        }
     else
 {
 System.out.println("The stack is not empty");
        }
    }
 }
```

Output:



Output - Stack (run)

```
run:
Inserting 1
Inserting 2
Removing 2
Removing 1
Inserting 3
The top element is 3
The stack size is 1
Removing 3
The stack is empty
BUILD SUCCESSFUL (total time: 1 second)
```

B) QUEUE IMPLEMENTATION

Program:

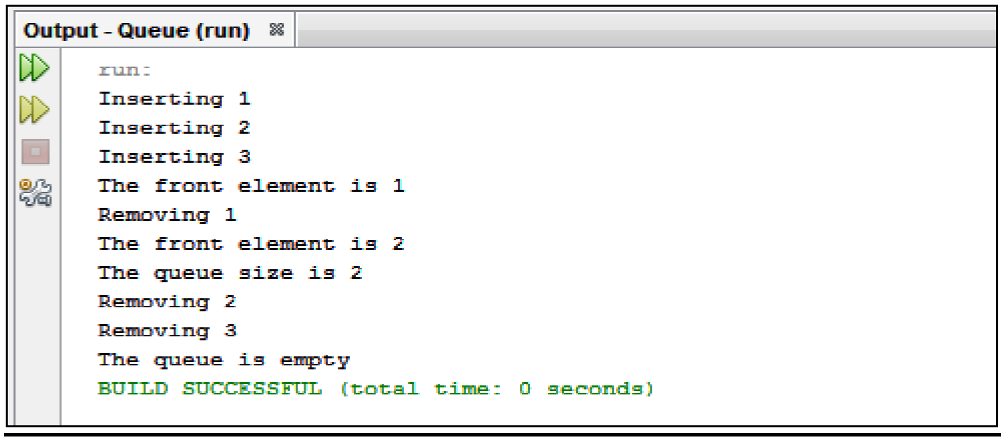
```
// A class to represent a queue
class Queue
{
private int[] arr;      // array to store queue elements
private int front;      // front points to the front element in the queue
private int rear;       // rear points to the last element in the queue
private int capacity;   // maximum capacity of the queue
private int count;      // current size of the queue

   // Constructor to initialize a queue
Queue(int size)
   {
arr = new int[size];
capacity = size;
front = 0;
rear = -1;
count = 0;
   }
    // Utility function to dequeue the front element
public int dequeue()
   {
       // check for queue underflow
if (isEmpty())
       {
System.out.println("Underflow\nProgram Terminated");
System.exit(-1);
       }
int x = arr[front];
System.out.println("Removing " + x);
front = (front + 1) % capacity;
count--;
return x;
   }
    // Utility function to add an item to the queue
public void enqueue(int item)
   {
      // check for queue overflow
if (isFull())
      {
System.out.println("Overflow\nProgram Terminated");
System.exit(-1);
      }

System.out.println("Inserting " + item);
rear = (rear + 1) % capacity;
arr[rear] = item;
count++;
   }
    // Utility function to return the front element of the queue
public int peek()
   {
if (isEmpty())
      {
System.out.println("Underflow\nProgram Terminated");
System.exit(-1);
      }
return arr[front];
   }
    // Utility function to return the size of the queue
public int size() {
return count;
   }
    // Utility function to check if the queue is empty or not
public boolean isEmpty() {
return (size() == 0);
   }
```

```
 // Utility function to check if the queue is full or not
public boolean isFull() {
return (size() == capacity);
   }
}
 class Main
{
public static void main (String[] args)
   {
        // create a queue of capacity 5
Queue q = new Queue(5);
q.enqueue(1);
q.enqueue(2);
q.enqueue(3);
System.out.println("The front element is " + q.peek());
q.dequeue();
System.out.println("The front element is " + q.peek());
System.out.println("The queue size is " + q.size());
q.dequeue();
q.dequeue();
if (q.isEmpty())
 {
System.out.println("The queue is empty");
        }
        else
{
System.out.println("The queue is not empty");
        }    }              }
```

Output:



```
run:
Inserting 1
Inserting 2
Inserting 3
The front element is 1
Removing 1
The front element is 2
The queue size is 2
Removing 2
Removing 3
The queue is empty
BUILD SUCCESSFUL (total time: 0 seconds)
```

Expt. No: 3	EMPLOYEE SALARY CALCULATION USING INHERITANCE CONCEPTS
Date:	

Program

```
import java.util.Scanner;
public class EmployeeSalaryCalc
{
public static void main(Stringargs[])
{
      Scanner obj=newScanner(System.in);
      Programmer p=newProgrammer();
      System.out.println("Enter the basic pay of Programmer");
      p.getEmployeeDetails(obj.nextDouble());
      p.cal();
      AssistantProfessor ap=new AssistantProfessor();
      System.out.println("Enter the basic pay of Assistant Professor");
      ap.getEmployeeDetails(obj.nextDouble());
      ap.cal();
      AssociateProfessor asp=new AssociateProfessor();
      System.out.println("Enter the basic pay of Associate Professor");
      asp.getEmployeeDetails(obj.nextDouble());
      asp.cal();
      Professor prof=new Professor();
      System.out.println("Enter the basic pay ofProfessor");
      prof.getEmployeeDetails(obj.nextDouble());
      prof.cal();
   }
}
class Employee{
   String employeeName;
   int employeeID;
   Stringaddress;
   StringmailID;
```

```
 long mobileNumber;
 double da,hra,pf,sc,ns,gs;
 Scanner obj=new Scanner(System.in);
 void getEmployeeDetails()
 {
  System.out.println("Enter the Employee Name:");
  employeeName=obj.nextLine();
  System.out.println("Enter the Employee Address:");
  address=obj.nextLine();
  System.out.println("Enter the Employee Mail ID:");
  mailID=obj.nextLine();
  System.out.println("Enter the Employee ID:");
  employeeID=obj.nextInt();
  System.out.println("Enter the Employee Mobile Number:");
  mobileNumber=obj.nextLong();
}

void display()

{
  System.out.println("EmployeeName          :"+employeeName);
  System.out.println("EmployeeID            :"+employeeID);
  System.out.println("EmployeeAddress       :"+address);
  System.out.println("EmployeeMail ID       :"+mailID);
  System.out.println("Employee MobileNumber:"+mobileNumber);
  }
  }
  class Programmer extends Employee
  {
  double basicPay;
  public double getBasicPay()
  {
  return basicPay;
  }
```

```
public void setBasicPay(double basicPay)
 {
 this.basicPay = basicPay;
  }
  void getEmployeeDetails(double bp)
  {
  super.getEmployeeDetails();
  setBasicPay(bp);
  }
  void cal(){ da=getBasicPay()*97/100.0;
  hra=getBasicPay()*10/100.0;
  pf=getBasicPay()*12/100.0;
  sc=getBasicPay()*1/100.0;
  gs=getBasicPay()+da+hra+pf+sc;
  ns=gs-pf-sc;
  display();
  }
  void display()
  {
  super.display();
  System.out.println("Employee Gross Salary:"+gs);
  System.out.println("Employee Net Salary:"+ns);
  }
  }
   class AssistantProfessor extendsEmployee
   {
   double basicPay;
   public double
  getBasicPay()
    {
   return basicPay;
   }
   public void setBasicPay(double basicPay)
```

```
{

 this.basicPay = basicPay;
}
void getEmployeeDetails(double bp)
{
 super.getEmployeeDetails();
 setBasicPay(bp);
 }
 void cal(){
 da=getBasicPay()*110/100.0;
 hra=getBasicPay()*20/100.0;
 pf=getBasicPay()*12/100.0;
 sc=getBasicPay()*5/100.0;
 gs=getBasicPay()+da+hra+pf+
 sc; ns=gs-pf-sc;
 display();
 }
 void display()
 {
  super.display();
 System.out.println("Employee Gross Salary:"+gs);
 System.out.println("Employee Net Salary:"+ns);
 }
 }
 class AssociateProfessor extends Employee
 {
 double basicPay;
 public double
 getBasicPay()
  {
 return basicPay;
 }
 public void setBasicPay(double basicPay)
 {
```

```
 this.basicPay = basicPay;
 }
 void getEmployeeDetails(double bp)
 {
 super.getEmployeeDetails();
 setBasicPay(bp);
 }
 void cal()
 {
 da=getBasicPay()*130/100.0;
 hra=getBasicPay()*30/100.0;
 pf=getBasicPay()*12/100.0;
 sc=getBasicPay()*10/100.0;
 gs=getBasicPay()+da+hra+pf+
 sc; ns=gs-pf-sc;
 display();
 }
void display()
{
super.display();
System.out.println("Employee Gross Salary:"+gs);
System.out.println("Employee Net Salary:"+ns);
}
}
class Professor extends Employee
{
double basicPay;
public double
getBasicPay()
 {
 return basicPay;
}
public void setBasicPay(double basicPay)
```

```
 {
 this.basicPay = basicPay;
}
void getEmployeeDetails(double bp)
{
super.getEmployeeDetails();
setBasicPay(bp);
}
void cal()
{
da=getBasicPay()*140/100.0;
hra=getBasicPay()*40/100.0;
pf=getBasicPay()*12/100.0;
sc=getBasicPay()*15/100.0;
gs=getBasicPay()+da+hra+pf+sc;
ns=gs-pf-sc;
display();
}
void display(){
super.display();
System.out.println("Employee Gross Salary:"+gs);
System.out.println("Employee Net Salary:"+ns);
}
}
```

Output:

```
Enter the basic pay of Programmer
15000
Enter the Employee Name:
ram
Enter the Employee Address:
56 Ganga Street
Enter the Employee Mail
ID: ram@gmail.com
Enter the Employee ID:
101
Enter the Employee Mobile
Number: 9994117284
EmployeeName       :ram
EmployeeID        101

EmployeeAddress    :56 Ganga Street
EmployeeMail ID
                   :ram@gmail.com
Employee
MobileNumber:9994117284
Employee Gross Salary:33000.0
Employee Net Salary:31050.0
Enter the basic pay of Assistant Professor
20000
Enter the Employee Name:
vinu
Enter the Employee Address:
75 public office road
Enter the Employee Mail
ID: vinu@gmail.com
Enter the Employee ID:
201
```

```
Enter the Employee Mobile
Number: 9842321130
EmployeeName        :vinu
Employee ID         201
EmployeeAddress     :75 public office road
EmployeeMail ID     :vinu@gmail.com
Employee Mobile Number:9842321130
Employee Gross Salary:49400.0
Employee Net Salary:46000.0
```

Enter the basic pay of Associate Professor

```
30000
Enter the Employee Name:
krish
Enter the Employee Address:
25 neela east street
Enter the Employee Mail
ID: krish@gmail.com
Enter the Employee ID:
301
Enter the Employee Mobile
Number: 9578621131
EmployeeName        :krish
EmployeeID          301
EmployeeAddress     :25 neela east street
EmployeeMail ID     :krish@gmail.com
Employee Mobile Number:9578621131
Employee Gross Salary:84600.0
Employee Net Salary:78000.0
```

```
Enter the basic pay of Professor
40000
Enter the Employee
Name: vinayagam
Enter the Employee Address:
100 Nehru Street
Enter the Employee Mail
ID: vinayagam@gmail.com
Enter the Employee ID:
401
Enter the Employee Mobile
Number: 7904923391
EmployeeName:vinayagam
Employee ID       401
EmployeeAddress    :100 Nehru Street
EmployeeMail ID:vinayagam@gmail.com
Employee Mobile Number:7904923391
Employee Gross Salary:122800.0
Employee Net Salary:112000.0
```

Expt. No: 4	ABSTRACT CLASS
Date:	

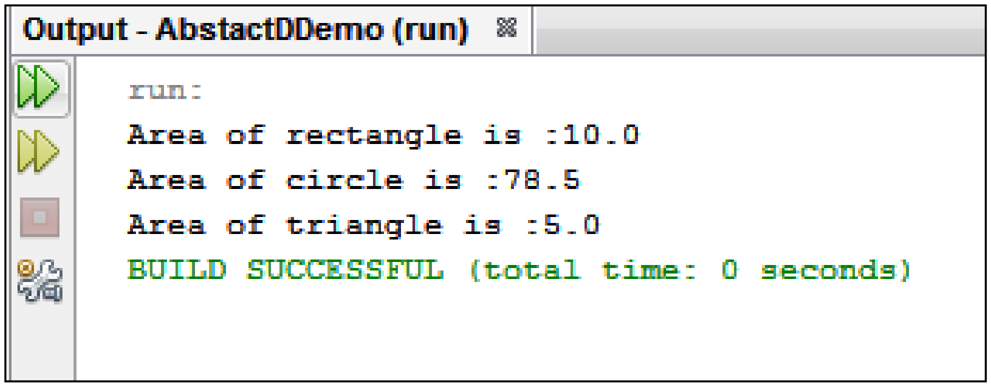
Program

```
import java.util.*;
abstract class shape
{
int x,y;
abstract void area(double x,double y);
}
class Rectangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle extends shape
{
void area(double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
}
}
class Triangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of triangle is :"+(0.5*x*y));
}
}
public class AbstactDDemo
{
```

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

{
Rectangle r=new Rectangle();
r.area(2,5);
Circle c=new Circle();
c.area(5,5);
Triangle t=new Triangle();
t.area(2,5);
}
}
```

Output:



Output - AbstactDDemo (run)

```
run:
Area of rectangle is :10.0
Area of circle is :78.5
Area of triangle is :5.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

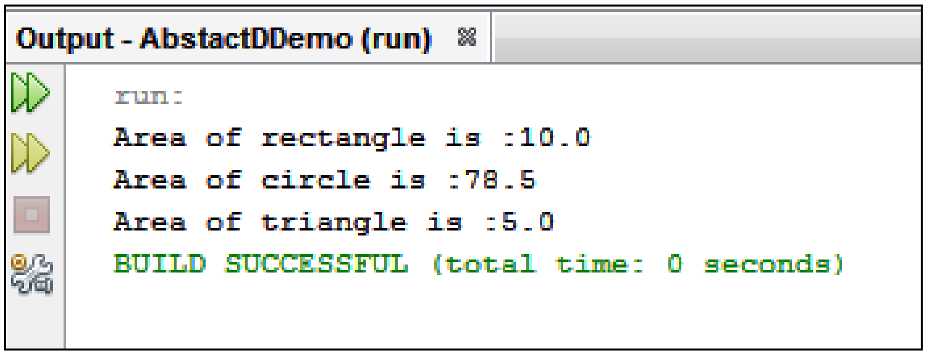
Expt. No: 5	**INTERFACE**
Date:	

Program:

```
interface Shape
{
void input();
void area();
}
class Circle implements Shape
{
int r = 0;
double pi = 3.14, ar = 0;
@Override
public void input()
  {
    r = 5;
  }
@Override
public void area()
  {
ar = pi * r * r;
System.out.println("Area of circle:"+ar);
  }
}
```

```
class Rectangle extends Circle
{
int l = 0, b = 0;
double ar;
public void input()
 {
super.input();
      l = 6;
      b = 4;
   }
public void area()
   {
super.area();
ar = l * b;
System.out.println("Area of rectangle:"+ar);
   }
}
public class Demo
{
public static void main(String[] args)
{
      Rectangle obj = new Rectangle();
obj.input();
obj.area();
   }
}
```

Output:



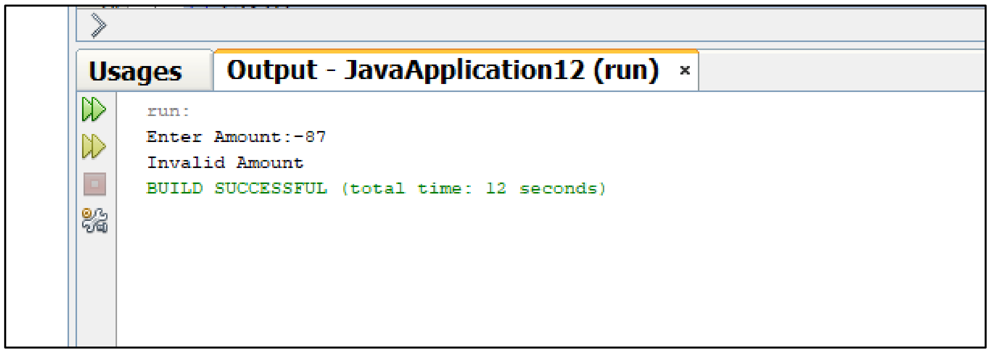
```
run:
Area of rectangle is :10.0
Area of circle is :78.5
Area of triangle is :5.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Expt. No: 6	
Date:	USER DEFINED EXCEPTION HANDLING

Program:

```
package javaapplication12;
import java.util.Scanner;
class NegativeAmtException extends Exception
        {
        String msg;
        NegativeAmtException(String msg)
        {
            this.msg=msg;
        }
        public String toString()
        {
        return msg;
        }
        }
        public class userdefined
        {
        public static void main(String[] args)
        {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter Amount:");
        int a=s.nextInt();
        try
        {
        if(a<0)
         {
        throw new NegativeAmtException("Invalid Amount");
        }
System.out.println("Amount Deposited");
}
catch(NegativeAmtException e)
{
System.out.println(e);
}
}
}
```

Output:



```
run:
Enter Amount:-87
Invalid Amount
BUILD SUCCESSFUL (total time: 12 seconds)
```

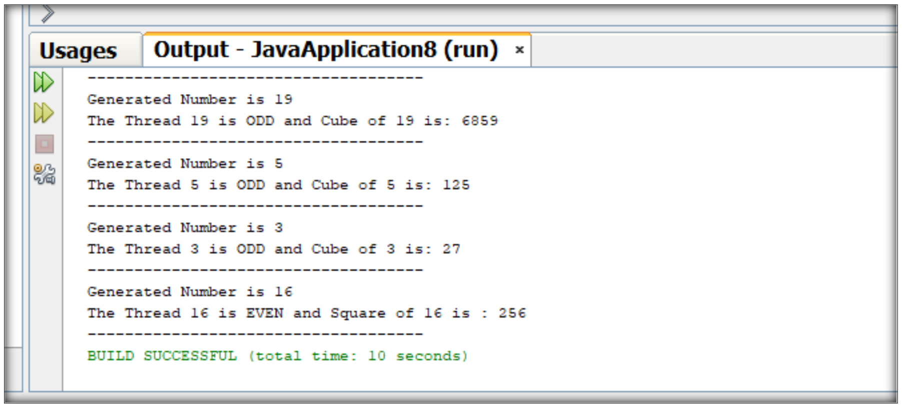
Expt. No: 7	MULTI THREADED APPLICATION
Date:	

Program:

```
import java.util.*;
class EvenNum implements Runnable
{
public int a;
public EvenNum(int a)
{
this.a = a;
}
public void run()
{
System.out.println("The Thread "+ a +" is EVEN and Square of " + a + " is : " + a * a);
 }
 }
class OddNum implements Runnable
{
public int a;
public OddNum(int a)
{
      this.a = a;
   }
public void run()
 {
System.out.println("The Thread "+ a +" is ODD and Cube of " + a + " is: " + a * a * a);
   }
}
class RandomNumGenerator extends Thread
{
public void run()
{
```

```
int n = 0;
Random rand = new Random();
 try
{
for (int i = 0; i < 10; i++)
{
 n = rand.nextInt(20);
System.out.println("Generated Number is " + n);
if (n % 2 == 0)
{
        Thread thread1 = new Thread(new EvenNum(n));
thread1.start();
     }
else
 {   Thread thread2 = new Thread(new OddNum(n));
thread2.start();
 }
Thread.sleep(1000);
System.out.println("------------------------------------");
}
}
  catch (Exception ex)
{
System.out.println(ex.getMessage());
}
}
}
public class MultiThreadRandOddEven
{
public static void main(String[] args)
{
   RandomNumGenerator rand_num = new RandomNumGenerator();
rand_num.start();
}}
```

Output:



```
------------------------------------
Generated Number is 19
The Thread 19 is ODD and Cube of 19 is: 6859
------------------------------------
Generated Number is 5
The Thread 5 is ODD and Cube of 5 is: 125
------------------------------------
Generated Number is 3
The Thread 3 is ODD and Cube of 3 is: 27
------------------------------------
Generated Number is 16
The Thread 16 is EVEN and Square of 16 is : 256
------------------------------------
BUILD SUCCESSFUL (total time: 10 seconds)
```

Expt. No: 8	
Date:	**FILE OPERATIONS**

Program:

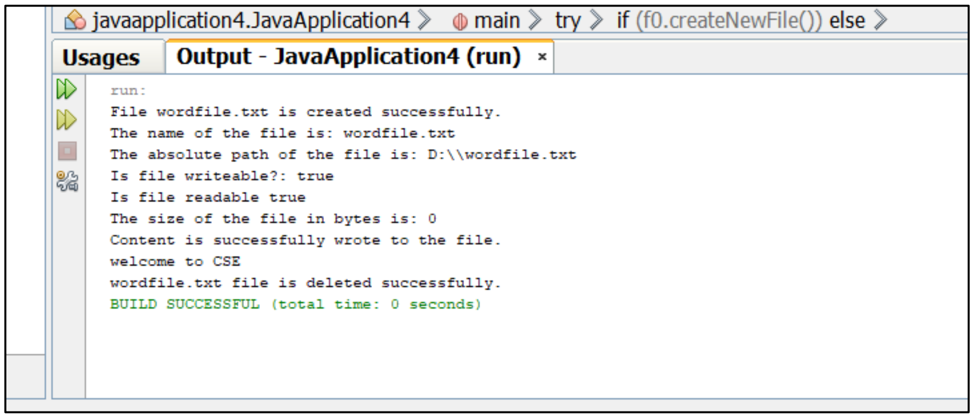
```
package javaapplication4;
import java.io.IOException;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.util.Scanner;
public class JavaApplication4
{
public static void main(String args[])
{
      File f0 = new File("D:wordfile.txt");
 try
{
if (f0.createNewFile())
  {
System.out.println("File " + f0.getName() + " is created successfully.");
 }
  else
  {
System.out.println("File is already exist in the directory.");
 }
   }
catch (IOException exception)
{
System.out.println("An unexpected error is occurred.");
exception.printStackTrace();
}
```

```
if (f0.exists())
{
System.out.println("The name of the file is: " + f0.getName());
System.out.println("The absolute path of the file is: " + f0.getAbsolutePath());
System.out.println("Is file writeable?: " + f0.canWrite());
System.out.println("Is file readable " + f0.canRead());
System.out.println("The size of the file in bytes is: " + f0.length());
}
 else
{
System.out.println("The file does not exist.");
}
try
{
FileWriter fwrite = new FileWriter("D:wordfile.txt");
fwrite.write("welcome to CSE");

fwrite.close();
System.out.println("Content is successfully wrote to the file.");
}
catch(IOException e)
{
System.out.println("Unexpected error occurred");
e.printStackTrace();
}
try
{
 File f1 = new File("D:wordfile.txt");
  Scanner dataReader = new Scanner(f1);
  while (dataReader.hasNextLine())
{
 String fileData = dataReader.nextLine();
System.out.println(fileData);
}
dataReader.close();
}
catch (FileNotFoundException exception)
{
System.out.println("Unexcpected error occurred!");
exception.printStackTrace();
}
try
{
File f1 = new File("D:FileOperationExample.txt");
Scanner dataReader = new Scanner(f1);
while (dataReader.hasNextLine())
{
String fileData = dataReader.nextLine();
System.out.println(fileData);
}

dataReader.close();
}
catch (FileNotFoundException exception)
 {
System.out.println("Unexcpected error occurred!");
exception.printStackTrace();
}
if (f0.delete())
   {
System.out.println(f0.getName()+ " file is deleted successfully.");
    }
else
{
System.out.println("Unexpected error found in deletion of the file.");
}      }  }
```

Output:


javaapplication4.JavaApplication4 > main > try > if (f0.createNewFile()) else >
Usages
Output - JavaApplication4 (run)

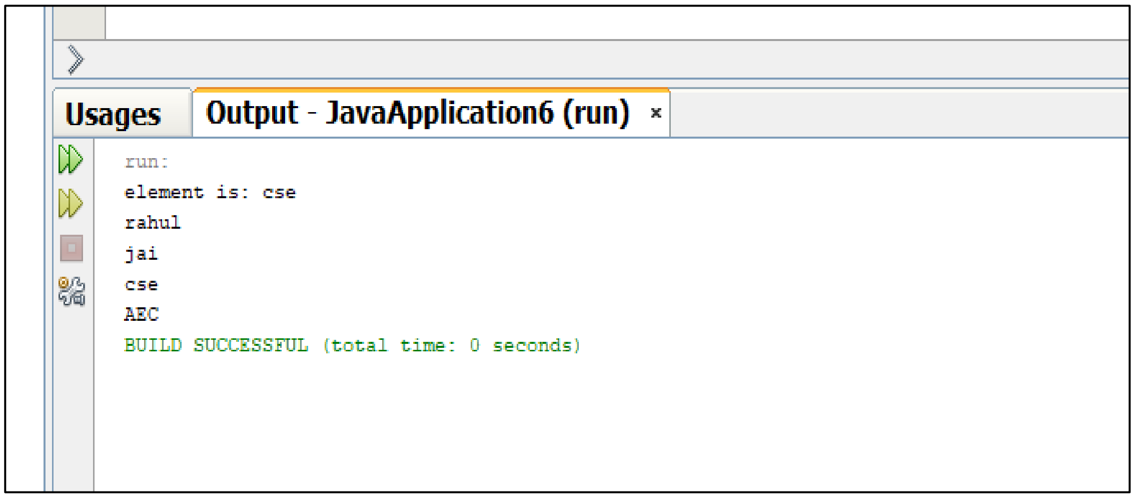
```
run:
File wordfile.txt is created successfully.
The name of the file is: wordfile.txt
The absolute path of the file is: D:\\wordfile.txt
Is file writeable?: true
Is file readable true
The size of the file in bytes is: 0
Content is successfully wrote to the file.
welcome to CSE
wordfile.txt file is deleted successfully.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Expt. No: 9	GENERICS CLASSES
Date:	

Program:

```
package javaapplication6;
import java.util.*;
import java.util.Iterator;
public class JavaApplication6
{
public static void main(String args[]){
ArrayList<String> list=new ArrayList<String>();
list.add("rahul");
list.add("jai");
list.add("cse");
list.add("AEC");
//list.add(32);//compile time error
String s=list.get(2);//type casting is not required
System.out.println("element is: "+s);
Iterator<String> itr=list.iterator();
while(itr.hasNext()){
System.out.println(itr.next());
}
}
}
```

Output:



```
run:
element is: cse
rahul
jai
cse
AEC
BUILD SUCCESSFUL (total time: 0 seconds)
```

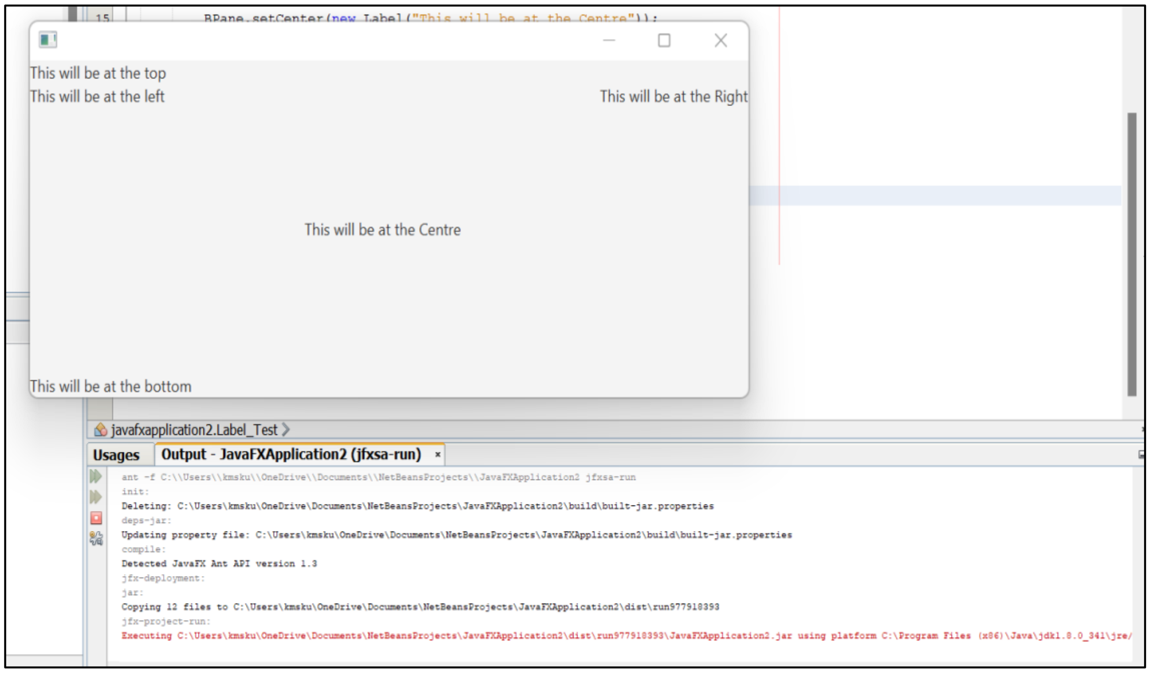
Expt. No: 10	JAVAFX CONTROLS
Date:	

A)LAYOUT

Program:

```
package javafxapplication2;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.*;
import javafx.stage.Stage;
public class Label_Test extends Application
{
@Override
public void start(Stage primaryStage) throws Exception
{
BorderPane BPane = new BorderPane();
BPane.setTop(new Label("This will be at the top"));
BPane.setLeft(new Label("This will be at the left"));
BPane.setRight(new Label("This will be at the Right"));
BPane.setCenter(new Label("This will be at the Centre"));
BPane.setBottom(new Label("This will be at the bottom"));
Scene scene = new Scene(BPane,600,400);
primaryStage.setScene(scene);
primaryStage.show();
}
public static void main(String[] args)
{
launch(args);
}
}
```

Output:



```
BPane.setCenter(new Label("This will be at the Centre"));
```

```
This will be at the top
This will be at the left                                   This will be at the Right

                    This will be at the Centre

This will be at the bottom
```

javafxapplication2.Label_Test
Usages
Output - JavaFXApplication2 (jfxsa-run)

```
ant -f C:\\Users\\kmsku\\OneDrive\\Documents\\NetBeansProjects\\JavaFXApplication2 jfxsa-run
init:
Deleting: C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication2\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication2\build\built-jar.properties
compile:
Detected JavaFX Ant API version 1.3
jfx-deployment:
jar:
Copying 12 files to C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication2\dist\run977918393
jfx-project-run:
Executing C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication2\dist\run977918393\JavaFXApplication2.jar using platform C:\Program Files (x86)\Java\jdk1.8.0_341\jre/
```

B)MENU

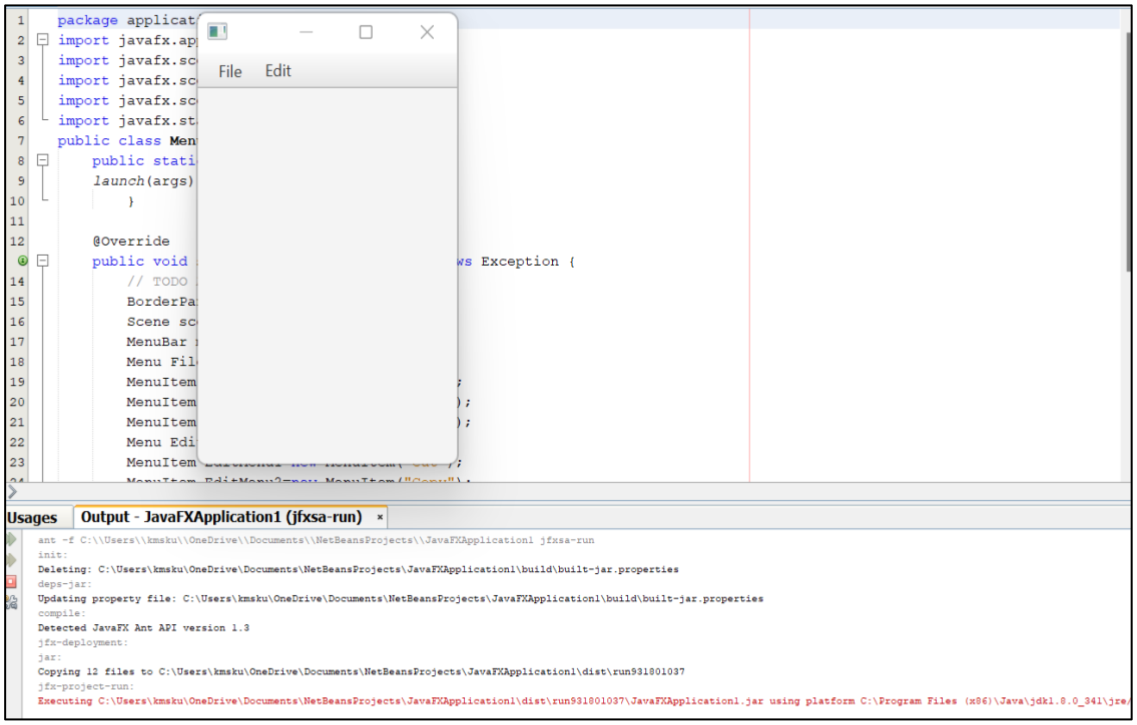
Program:

```
package application;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;
public class MenuExample extends Application
{
public static void main(String[] args)
 {
launch(args);
 }
   @Override
public void start(Stage primaryStage) throws Exception
 {

                                        // TODO Auto-generated method stub
 BorderPane root = new BorderPane();
 Scene scene = new Scene(root,200,300);
MenuBar menubar = new MenuBar();
Menu FileMenu = new Menu("File");
MenuItem filemenu1=new MenuItem("new");
MenuItem filemenu2=new MenuItem("Save");
    MenuItem filemenu3=new MenuItem("Exit");
    Menu EditMenu=new Menu("Edit");
   MenuItem EditMenu1=new MenuItem("Cut");
   MenuItem EditMenu2=new MenuItem("Copy");
   MenuItem EditMenu3=new MenuItem("Paste");
   EditMenu.getItems().addAll(EditMenu1,EditMenu2,EditMenu3);
   root.setTop(menubar);
```

```
        FileMenu.getItems().addAll(filemenu1,filemenu2,filemenu3);
        menubar.getMenus().addAll(FileMenu,EditMenu);
        primaryStage.setScene(scene);
        primaryStage.show();
        }
    }
```

Output:



```
package applicat
import javafx.ap
import javafx.sc
import javafx.sc
import javafx.sc
import javafx.st
public class Men
    public stati
    launch(args)
        }

    @Override
    public void                                  ws Exception {
        // TODO
        BorderPa
        Scene sc
        MenuBar
        Menu Fil
        MenuItem                                 ;
        MenuItem                                 );
        MenuItem                                 );
        Menu Edi
        MenuItem                                 ;
```

```
ant -f C:\\Users\\kmsku\\OneDrive\\Documents\\NetBeansProjects\\JavaFXApplication1 jfxsa-run
init:
Deleting: C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication1\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication1\build\built-jar.properties
compile:
Detected JavaFX Ant API version 1.3
jfx-deployment:
jar:
Copying 12 files to C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication1\dist\run931801037
jfx-project-run:
Executing C:\Users\kmsku\OneDrive\Documents\NetBeansProjects\JavaFXApplication1\dist\run931801037\JavaFXApplication1.jar using platform C:\Program Files (x86)\Java\jdk1.8.0_341\jre/
```

Expt. No:11	MINI PROJECT
Date:	

Program:

```
package javaapplication11;
import java.awt.*;
import java.awt.event.*;
public class MyCalculator extends Frame
{
public boolean setClear=true;
double number, memValue;
char op;
String digitButtonText[] = {"7", "8", "9", "4", "5", "6", "1", "2", "3", "0", "+/-", "." };
String operatorButtonText[] = {"/", "sqrt", "*", "%", "-", "1/X", "+", "=" };
String memoryButtonText[] = {"MC", "MR", "MS", "M+" };
String specialButtonText[] = {"Backspc", "C", "CE" };
MyDigitButton digitButton[]=new MyDigitButton[digitButtonText.length];
MyOperatorButton operatorButton[]=new MyOperatorButton[operatorButtonText.length];
MyMemoryButton memoryButton[]=new MyMemoryButton[memoryButtonText.length];
MySpecialButton specialButton[]=new MySpecialButton[specialButtonText.length];
Label displayLabel=new Label("0",Label.RIGHT);
Label memLabel=new Label(" ",Label.RIGHT);
final int FRAME_WIDTH=325,FRAME_HEIGHT=325;
final int HEIGHT=30, WIDTH=30, H_SPACE=10,V_SPACE=10;
final int TOPX=30, TOPY=50;

MyCalculator(String frameText)
{
super(frameText);
int tempX=TOPX, y=TOPY;
displayLabel.setBounds(tempX,y,240,HEIGHT);
displayLabel.setBackground(Color.BLUE);
displayLabel.setForeground(Color.WHITE);
```

```
add(displayLabel);
memLabel.setBounds(TOPX,  TOPY+HEIGHT+ V_SPACE,WIDTH, HEIGHT);
add(memLabel);
tempX=TOPX;
y=TOPY+2*(HEIGHT+V_SPACE);
for(int i=0; i<memoryButton.length; i++)
{
memoryButton[i]=new MyMemoryButton(tempX,y,WIDTH,HEIGHT,memoryButtonText[i], this);
memoryButton[i].setForeground(Color.RED);
y+=HEIGHT+V_SPACE;
}
tempX=TOPX+1*(WIDTH+H_SPACE); y=TOPY+1*(HEIGHT+V_SPACE);
for(int i=0;i<specialButton.length;i++)
{
specialButton[i]=new MySpecialButton(tempX,y,WIDTH*2,HEIGHT,specialButtonText[i], this);
specialButton[i].setForeground(Color.RED);
tempX=tempX+2*WIDTH+H_SPACE;
}
int digitX=TOPX+WIDTH+H_SPACE;
int digitY=TOPY+2*(HEIGHT+V_SPACE);
tempX=digitX;  y=digitY;
for(int i=0;i<digitButton.length;i++)
{
digitButton[i]=new MyDigitButton(tempX,y,WIDTH,HEIGHT,digitButtonText[i], this);
digitButton[i].setForeground(Color.BLUE);
tempX+=WIDTH+H_SPACE;
if((i+1)%3==0){tempX=digitX; y+=HEIGHT+V_SPACE;}
}
int opsX=digitX+2*(WIDTH+H_SPACE)+H_SPACE;
int opsY=digitY;
tempX=opsX;  y=opsY;
for(int i=0;i<operatorButton.length;i++)
{
tempX+=WIDTH+H_SPACE;
operatorButton[i]=new MyOperatorButton(tempX,y,WIDTH,HEIGHT,operatorButtonText[i], this);
operatorButton[i].setForeground(Color.RED);
if((i+1)%2==0){tempX=opsX; y+=HEIGHT+V_SPACE;}
}
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent ev)
{
System.exit(0);
}
});
setLayout(null);
setSize(FRAME_WIDTH,FRAME_HEIGHT);
setVisible(true);
}
static String getFormattedText(double temp)
{
String resText=""+temp;
if(resText.lastIndexOf(".0")>0)
resText=resText.substring(0,resText.length()-2);
return resText;
}
public static void main(String []args)
{
new MyCalculator("Calculator - JavaTpoint");
}
}
class MyDigitButton extends Button implements ActionListener
{
MyCalculator cl;
MyDigitButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
```

```
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
static boolean isInString(String s, char ch)
{
for(int i=0; i<s.length();i++) if(s.charAt(i)==ch) return true;
return false;
}
public void actionPerformed(ActionEvent ev)
{
String tempText=((MyDigitButton)ev.getSource()).getLabel();
if(tempText.equals("."))
{
if(cl.setClear)
{
cl.displayLabel.setText("0.");cl.setClear=false;
}
else if(!isInString(cl.displayLabel.getText(),'.'))
cl.displayLabel.setText(cl.displayLabel.getText()+".");
return;
}
int index=0;
try
{
index=Integer.parseInt(tempText);
}
catch(NumberFormatException e)
{
return;
}
if (index==0 && cl.displayLabel.getText().equals("0")) return;
if(cl.setClear)
{
```

```
cl.displayLabel.setText(""+index);cl.setClear=false;
}
else
cl.displayLabel.setText(cl.displayLabel.getText()+index);
}
}
class MyOperatorButton extends Button implements ActionListener
{
MyCalculator cl;
MyOperatorButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
public void actionPerformed(ActionEvent ev)
{
String opText=((MyOperatorButton)ev.getSource()).getLabel();
cl.setClear=true;
double temp=Double.parseDouble(cl.displayLabel.getText());
if(opText.equals("1/x"))
{
try
{
double tempd=1/(double)temp;
cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));
}
catch(ArithmeticException excp)
{
cl.displayLabel.setText("Divide by 0.");
}
return;
```

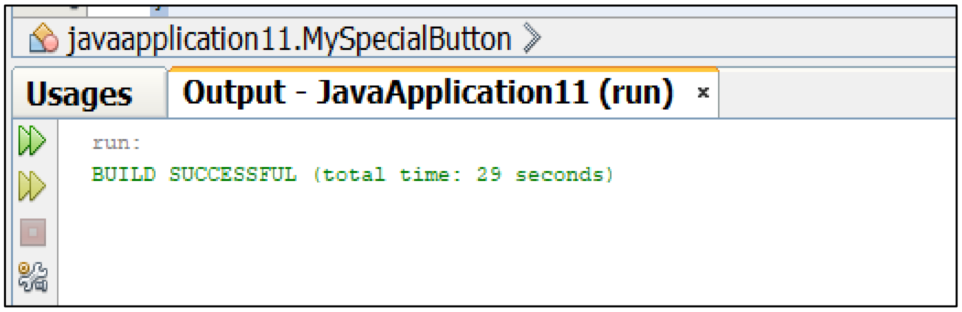
```
}
if(opText.equals("sqrt"))
{
try
{
double tempd=Math.sqrt(temp);
cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));
}
catch(ArithmeticException excp)
{
cl.displayLabel.setText("Divide by 0.");
}
return;
}
if(!opText.equals("="))
{
cl.number=temp;
cl.op=opText.charAt(0);
return;
}
switch(cl.op)
{
case '+':
temp+=cl.number;break;
case '-':
temp=cl.number-temp;break;
case '*':
temp*=cl.number;break;
case '%':
try
{
temp=cl.number%temp;
}
```

```
catch(ArithmeticException excp)
{
cl.displayLabel.setText("Divide by 0.");
return;
}
break;
case '/':
try
{
temp=cl.number/temp;
}
catch(ArithmeticException excp)
{
cl.displayLabel.setText("Divide by 0.");
 return;
}
break;
}//switch
cl.displayLabel.setText(MyCalculator.getFormattedText(temp));
}
}
class MyMemoryButton extends Button implements ActionListener
{
MyCalculator cl;
MyMemoryButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
public void actionPerformed(ActionEvent ev)
{
```

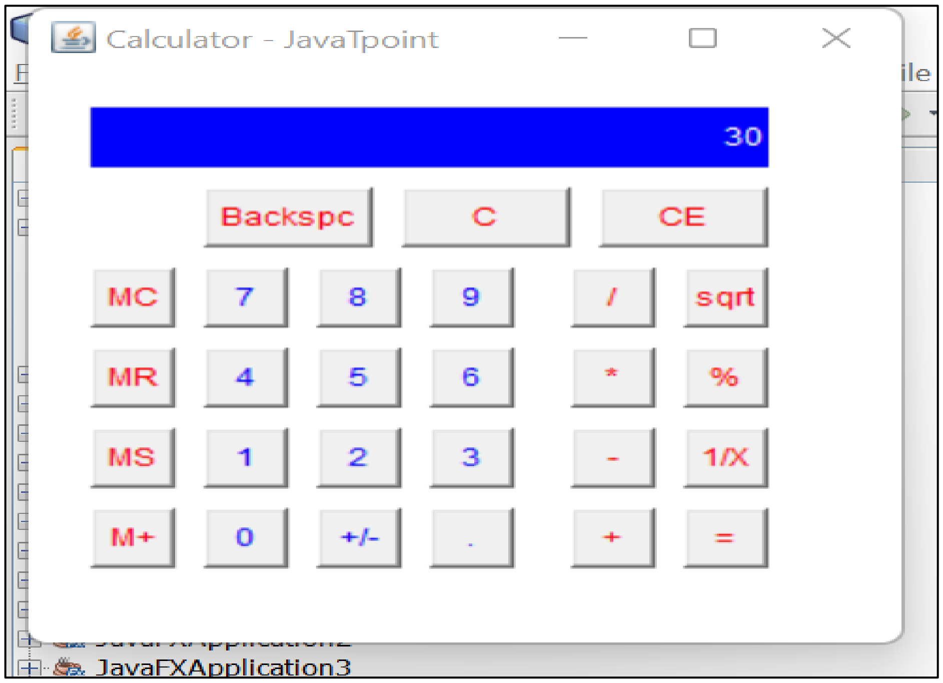
```
char memop=((MyMemoryButton)ev.getSource()).getLabel().charAt(1);
cl.setClear=true;
double temp=Double.parseDouble(cl.displayLabel.getText());
switch(memop)
{
case 'C':
cl.memLabel.setText(" ");cl.memValue=0.0;break;
case 'R':
cl.displayLabel.setText(MyCalculator.getFormattedText(cl.memValue));break;
case 'S':
cl.memValue=0.0;
case '+':
cl.memValue+=Double.parseDouble(cl.displayLabel.getText());
if(cl.displayLabel.getText().equals("0") || cl.displayLabel.getText().equals("0.0")  )
cl.memLabel.setText(" ");
else
cl.memLabel.setText("M");
break;
}
}
}
class MySpecialButton extends Button implements ActionListener
{
MyCalculator cl;
MySpecialButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
static String backSpace(String s)
{
```

```
String Res="";
for(int i=0; i<s.length()-1; i++) Res+=s.charAt(i);
return Res;
}
public void actionPerformed(ActionEvent ev)
{
String opText=((MySpecialButton)ev.getSource()).getLabel();
if(opText.equals("Backspc"))
{
String tempText=backSpace(cl.displayLabel.getText());
if(tempText.equals(""))
cl.displayLabel.setText("0");
else
cl.displayLabel.setText(tempText);
return;
}
if(opText.equals("C"))
{
cl.number=0.0; cl.op=' '; cl.memValue=0.0;
cl.memLabel.setText(" ");
}
cl.displayLabel.setText("0");cl.setClear=true;
}
}
```

Output:


javaapplication11.MySpecialButton
Usages
Output - JavaApplication11 (run)

```
run:
BUILD SUCCESSFUL (total time: 29 seconds)
```


Calculator - JavaTpoint
30
Backspc
C
CE
MC
7
8
9
/
sqrt
MR
4
5
6
*
%
MS
1
2
3
-
1/X
M+
0
+/-
.
+
=
JavaFXApplication3