

# **1. Basics of JAVA**

---

Thursday, 24th August, 2017

# Problem 1.1

Write a program to accept base and height of a right angled triangle from the user and compute its area and perimeter.

```
import java.util.*;
import static java.lang.Math.*;
public class Triangle
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter base length:");
        float b=sc.nextFloat();
        System.out.print("Enter height length:");
        float h=sc.nextFloat();
        if(b!=0 && h!=0)
        {
            System.out.print("Area="+((0.5*b*h)+" unit squared\nPerimeter="+((b+h+Math.hypot(b,h))+"
units"));
        }
        else
        {
            System.out.print("Triangle doesn't exist");
        }
    }
}
```

## Output

```
//Test case #1
Enter base length:3.6
Enter height length:4.28
Area=7.704000173568716 unit squared
Perimeter=13.47270975372258 units

//Test case #2
Enter base length:3
Enter height length:0
Triangle doesn't exist
```

## Problem 1.2

Write a program to perform addition, subtraction, multiplication and division of two numbers using command line arguments.

```
class Command
{
    public static void main(String[] args)
    {
        int i=Integer.parseInt(args[0]);
        int j=Integer.parseInt(args[1]);
        System.out.print("Sum="+ (i+j) + "\nDifference=" + (i-j)
        + "\nProduct=" + (i*j) + "\nQuotient=" + (i/j));
    }
}
```

### Output

```
$ 129 3
Sum=132
Difference=126
Product=387
Quotient=43
```

## Problem 1.3

Write a program to calculate Gross salary by accepting basic salary from the user.

$DA = 50\%$  Of basic salary

$HRA = 25\%$  Of basic salary

Gross Salary =  $DA + HRA + \text{basic salary}$

```
import java.util.*;
class Salary
{
    public static void main(String args[])
    {
        /* Let basic salary=x
        then gross salary=basic salary+da+hra=x+0.5x+0.25x=1.75*x */
        Scanner sc=new scanner(System.in);
        System.out.print("Enter your salary:");
        int s=sc.nextInt();
        System.out.print("Gross Salary="+ (1.75*s));
    }
}
```

## Output

```
//Test case #1
Enter your salary:14000
Gross Salary=24500

//Test case #2
Enter your salary:114578
Gross Salary=200515

//Test case #3
Enter your salary:0
Gross Salary=0
```

## **2. Control structures, operators and accepting data through keyboard**

---

Thursday, 31st August, 2017

## Problem 2.1

Write a menu driven program for the following case(s):

1. Check entered number is even or odd
2. Check entered number is Prime number or not.
3. Check entered number is palindrome number or not.

```
import java.util.*;
class Numbers
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter a number:");
        int n=sc.nextInt();
        System.out.print("MENU\n1.Even-Odd check\n2.Primality test\n3.Palindromic number
test\n");
        System.out.print("Enter your choice:");
        int ch=sc.nextInt();
        switch(ch)
        {
            case 1: if(n%2==0)
                    {
                        System.out.print("Even");
                    }
                    else
                    {
                        System.out.print("Odd");
                    }
                    break;
            case 2: if(n!=1)
                    {
                        int f=0;
                        for(int i=2;i<n;i++)
                        {
                            if(n%i==0)
                            {
                                f=1;
                                break;
                            }
                        }
                        if(f==0)
                        {
                            System.out.print("Prime Number");
                        }
                    }
        }
```

```

        else
        {
            System.out.print("Composite number");
        }
    }
    else
    {
        System.out.print("Neither prime nor composite");
    }
    break;
case 3: int tn=n,rev=0;
while(n>0)
{
    rev=rev*10+(n%10);
    n/=10;
}
if(tn==rev)
{
    System.out.print("Palindromic number");
}
else
{
    System.out.print("Not a palindromic number");
}
break;
default: System.out.print("Invalid Choice");
}
}
}
}

```

## Output

```

//Test case #1
Enter a number:919
MENU
1.Even-Odd check
2.Primality test
3.Palindromic number test
Enter your choice:1
Odd

//Test case #2
Enter a number:919
MENU
1.Even-Odd check
2.Primality test
3.Palindromic number test
Enter your choice:2
Prime Number

```

```

//Test case #3
Enter a number:919
MENU
1.Even-Odd check
2.Primality test
3.Palindromic number test
Enter your choice:3
Palindromic number

//Test case #4
Enter a number:1
MENU
1.Even-Odd check
2.Primality test
3.Palindromic number test
Enter your choice:2
Neither prime nor composite

//Test case #5
Enter a number:25
MENU
1.Even-Odd check
2.Primality test
3.Palindromic number test
Enter your choice:4
Invalid Choice

```

## Problem 2.2

Write a program to display Truth Table using logical operators.

```

//Truth table for AND gate only
class Truthtable
{
    public static void main(String[] args)
    {
        boolean a=true,b=true;
        System.out.println("TRUTH TABLE FOR AND GATE");
        for(int i=1;i<=2;i++)
        {
            for(int j=1;j<=2;j++)
            {
                System.out.println(a+" AND "+b+" = "+(a&b));
                b=!b;
            }
            a=!a;
        }
    }
}

```



```
}  
}  
}
```

## Output

```
TRUTH TABLE FOR AND GATE  
true AND true = true  
true AND false = false  
false AND true = false  
false AND false = false
```

## Problem 2.3

Write a program to accept 3 numbers and find the largest and smallest no using conditional operator.

```
import java.util.*;  
class Largest  
{  
    public static void main(String[] args)  
    {  
        Scanner sc=new Scanner(System.in);  
        System.out.print("Enter first number:");  
        int x=sc.nextInt();  
        System.out.print("Enter second number:");  
        int y=sc.nextInt();  
        System.out.print("Enter third number:");  
        int z=sc.nextInt();  
        int max=(x>y)?((x>z)?x:z):((y>z)?y:z);  
        if(x!=y && y!=z)  
        {  
            System.out.print("Largest number is "+max);  
        }  
        else  
        {  
            System.out.print("Result inconclusive");  
        }  
    }  
}
```

## Output

```
//Test case #1
Enter first number:5
Enter second number:9
Enter third number:7
Largest number is 9

//Test case #2
Enter first number:3
Enter second number:2
Enter third number:2
Result inconclusive
```

## Problem 2.4

Write a program to display patterns

a) 1

2 1 2

3 2 1 2 3

b) Pascal Triangle

c) Floyd's Triangle

```
import java.util.*;
class Patterns
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter height for pattern(s):");
        int n=sc.nextInt();
        System.out.println("Regular upright triangle");
        for(int i=1;i<=n;i++)
        {
            for(int sp=1;sp<=n-i;sp++)
            {
                System.out.print(" ");
            }
        }
    }
}
```

```

        for(int j=1;j<=i;j++)
        {
            System.out.print(j);
        }
        for(int j=i-1;j>=1;j--)
        {
            System.out.print(j);
        }
        System.out.print("\n");
    }
    System.out.println();
    System.out.println("Pascal's Triangle");
    int number=1;
    for(int i=0;i<n;i++)
    {
        for(int k=n; k>i; k--)
        {
            System.out.print(" ");
        }
        number = 1;
        for(int j=0;j<=i;j++)
        {
            System.out.print(number+ " ");
            number = number * (i - j) / (j + 1);
        }
        System.out.print("\n");
    }
    System.out.println();
    System.out.println("Floyd's Triangle");
    int k=1;
    for(int i=1; i<=n; i++)
    {
        for(int j=1; j<=i; j++, k++)
        {
            System.out.print(k+ " ");
        }
        System.out.print("\n");
    }
}
}

```

## Output

```

Enter height for pattern(s):5
Regular upright triangle
 1
121
12321
1234321
123454321

```

Pascal's Triangle

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Floyd's Triangle

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

## Problem 2.5

Program(s) to demonstrate use of labeled break and labeled continue statement.

```
class BreakwithLabel
{
    public static void main(String[] args)
    {
        int[][] A = {{ 32, 87, 3, 589 },{ 12, 1076, 2000, 8 },{ 622, 127, 77, 955 }};
        int x = 12,i,j=0;
        boolean flag = false;
        search:
        for (i = 0; i < A.length; i++)
        {
            for (j = 0; j < A[i].length;j++)
            {
                if (A[i][j] == x)
                {
                    flag = true;
                    break search;
                }
            }
        }

        if (flag)
        {
            System.out.println("Found "+x+" at "+"("+i+", "+j+")");
        }
        else
        {
            System.out.println(x + " not in the array");
        }
    }
}
```

## Output

Found 12 at (1,0)

```
class ContinuewithLabel
{
    public static void main(String[] args)
    {
        String S="Look for a substring in me";
        String s="sub";
        boolean flag = false;
        int max = S.length()-s.length();
        test:
        for (int i = 0; i <= max; i++)
        {
            int n = s.length(),j=i,k=0;
            while(n--!=0)
            {
                if (S.charAt(j++)!=s.charAt(k++))
                {
                    continue test;
                }
            }
            flag = true;
            break test;
        }
        System.out.println(flag?"Found it":"Didn't find it");
    }
}
```

## Output

Found it

## **3. Classes and objects**

---

Thursday, 14th September, 2017

## Problem 3

Define a class to represent a bank account. Include the following

Members: name of the depositor, account number, type of account balance amount in the account

Methods:

1. To assign initial values (read\_data)
2. To deposit an amount
3. To withdraw an amount after checking balance.
4. To display the name of the depositor and balance

```
import java.util.*;
class Program
{
    public static void main(String[] args)
    {
        Bank b=new Bank();
        b.read();
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter deposit amount:");
        double x=sc.nextDouble();
        b.deposit(x);
        System.out.print("Enter withdraw amount:");
        double y=sc.nextDouble();
        if(b.withdraw(y))
        {
            b.disp();
        }
        else
        {
            System.out.print("Insufficient Balance");
        }
    }
}

class Bank
{
    String name="";
    float bal=0;

    void read()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter your name:");
```

```

        name=sc.nextLine();
        System.out.print("Enter your balance:");
        bal=sc.nextFloat();
    }

    void disp()
    {
        System.out.println("Name-"+name);
        System.out.print("Balance="+bal);
    }

    void deposit(double x)
    {
        bal+=x;
    }

    boolean withdraw(double x)
    {
        if(x>bal)
        {
            return false;
        }
        else
        {
            bal-=x;
            return true;
        }
    }
}

```

## Output

```

//Test case #1
Enter your name:Pronoy Mandal
Enter your balance:10000
Enter deposit amount:25000
Enter withdraw amount:3500
Name-Pronoy Mandal
Balance=31500.0

//Test case #2
Enter your name:Pronoy Mandal
Enter your balance:10000
Enter deposit amount:2000
Enter withdraw amount:20000
Insufficient Balance

```



## **4. Constructors**

---

Thursday, 14th September, 2017

## Problem 4.1

Write a program to calculate volume of a box by using class.

Data members: height, width and depth

Member functions:

1. Function which returns volume of a box.
2. Use constructor to initialize box object.

```
class ProgramTwo
{
    public static void main(String[] args)
    {
        Box B=new Box(24.5,10.0,5.0);
        System.out.print("Volume="+B.volume());
    }
}

class Box
{
    double width=0.0,height=0.0,depth=0.0;
    Box(double w,double h,double d)
    {
        width=w;
        height=h;
        depth=d;
    }

    double volume()
    {
        return width*height*depth;
    }
}
```

## Output

```
Volume=1225.0
```

## Problem 4.2

Write a program to perform different arithmetic operations on complex numbers.

Data members: Real, imaginary.

Member Functions: to perform

1. Addition of two complex numbers
2. Subtraction of two complex numbers
3. Multiplication two complex numbers
4. Division of two complex numbers
5. Display Complex number
6. Constructor function to initialize Complex numbers.

```
class ComplexProgram
{
    public static void main(String[] args)
    {
        complex z1=new complex(2.0,3.0);
        complex z2=new complex(1.0,2.0);
        complex z3=new complex(0.0,0.0);
        z3.add(z1,z2);
        System.out.print("Addition: ");
        z3.disp();
        System.out.print("\n");
        z3.sub(z1,z2);
        System.out.print("Subtraction: ");
        z3.disp();
        System.out.print("\n");
        z3.prod(z1,z2);
        System.out.print("Product: ");
        z3.disp();
        System.out.print("\n");
        z3.div(z1,z2);
        System.out.print("Division: ");
        z3.disp();
    }
}
class complex
{
    double real,img;

    complex(double x,double y)
    {
        real=x;img=y;
    }
}
```

```

void add(complex z1,complex z2)
{
    real=z1.real+z2.real;
    img=z1.img+z2.img;
}

void sub(complex z1,complex z2)
{
    real=z1.real-z2.real;
    img=z1.img-z2.img;
}

void prod(complex z1,complex z2)
{
    real=(z1.real*z2.real-z1.img*z2.img);
    img=(z1.real*z2.img+z2.real*z1.img);
}

void div(complex z1,complex z2)
{
    real=(z1.real*z2.real+z1.img*z2.img)/(z2.real*z2.real+z2.img*z2.img);
    img=(z1.img*z2.real-z1.real*z2.img)/(z2.real*z2.real+z2.img*z2.img);
}

void disp()
{
    if(img>=0)
        System.out.print(real+" + "+img+"i");
    else
        System.out.print(real+" - "+(-img)+"i");
}
}

```

## Output

```

//Test Case #1
Addition: 3.0 + 5.0i
Subtraction: 1.0 + 1.0i
Product: -4.0 + 7.0i
Division: 1.6 - 0.2i

//Test Case #2
Addition: 3.0 + 8.0i
Subtraction: 1.0 - 2.0i
Product: -13.0 + 13.0i
Division: 0.6538461538461539 - 0.2692307692307692i

```

# **5. Method and Constructor Overloading**

---

Thursday, 28th September, 2017

# Problem 5.1

Write a program to find the area of various geometrical shapes by using method overloading.

```
class MethodOverload
{
    public static void main(String[] args)
    {
        Areas A=new Areas();
        System.out.print("For square of side 4.5 .....\\n");
        A.area(4.5);
        System.out.print("For rectangle of dimensions 4.5 X 3.5 .....\\n");
        A.area(4.5,3.5);
        System.out.print("For triangle of sides 2.5,3.5 and 4.5 .....\\n");
        A.area(2.5,3.5,4.5);
    }
}

class Areas
{
    void area(double x)
    {
        System.out.print("Area="+x*x+" units squared\\n");
    }

    void area(double x,double y)
    {
        System.out.print("Area="+x*y+" units squared\\n");
    }

    void area(double x,double y,double z)
    {
        double s=(x+y+z)/2;
        System.out.print("Area="+Math.sqrt(s*(s-x)*(s-y)*(s-z))+" units squared\\n");
    }
}
```

## Output

```
For square of side 4.5 .....
Area=20.25 units squared
For rectangle of dimensions 4.5 X 3.5 .....
Area=15.75 units squared
For triangle of sides 2.5,3.5 and 4.5 .....
Area=4.353070037341462 units squared
```

## Problem 5.2

Write a program to perform addition of  $n$  numbers. (Overload add method depending on parameters).

```
class AddOverload
{
    public static void main(String[] args)
    {
        Addition A=new Addition();
        int[] arr={2,5,6,1,2,4,8,9,0};
        System.out.println("Sum of an integer array-");
        A.add(arr);
        System.out.println("Sum of a double array-");
        double [] arr2={2.4,6.7,4.965,0.5123,4.126,2.319};
        A.add(arr2);
        String[] arr3={"My","name","is","Pronoy","Mandal"};
        System.out.println("Concatenation of a string array-");
        A.add(arr3);
    }
}

class Addition
{
    void add(int[] A)
    {
        int S=0;
        for(int i=0;i<A.length;i++)
        {
            S+=A[i];
        }
        System.out.println("Sum="+S);
    }

    void add(double[] A)
    {
        double S=0.0;
        for(int i=0;i<A.length;i++)
        {
            S+=A[i];
        }
        System.out.println("Sum="+S);
    }
}
```

```
void add(String[] A)
{
    String S="";
    for(int i=0;i<A.length;i++)
    {
        S+=(A[i]+" ");
    }
    System.out.println("Sum="+S);
}
```

## Output

```
Sum of an integer array-
Sum=37
Sum of a double array-
Sum=21.022299999999998
Concatenation of a string array-
Sum=My name is Pronoy Mandal
```



## **6. Overriding**

---

Monday, 16th October, 2017

# Problem 6

Write a program to implement Overriding

```
class Override
{
    public static void main(String[] args)
    {
        B obj1=new B();
        C obj2=new C();
        obj1.display();
        System.out.println();
        obj2.display();
    }
}

class A
{
    void display()
    {
        System.out.print("This is from A");
    }
}

class B extends A
{
    void display()
    {
        System.out.print("This is from B");
    }
}

class C extends A
{
    void display()
    {
        System.out.print("This is from C");
    }
}
```

## Output

```
This is from B
This is from C
```

# **7. Inheritance**

---

Thursday, 12th October, 2017

# Problem 7

Write a program to implement inheritance

```
import java.util.Scanner;

class Shape
{
    int l;
    void readData()
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter the dimension:");
        l = sc.nextInt();
    }
}

class Circle extends Shape
{
    void calculateArea()
    {
        System.out.print("Area="+3.142*l*l+" units squared");
    }
}

class Square extends Shape
{
    void calculateArea()
    {
        System.out.print("Area="+l*l+" units squared");
    }
}

class Main
{
    public static void main(String[] args)
    {
        Circle c=new Circle();
        System.out.println("-----Cirle-----");
        c.readData();
        c.calculateArea();
        System.out.println();
        Square s=new Square();
        System.out.println("-----Square-----");
        s.readData();
        s.calculateArea();
    }
}
```

## Output

```
-----Cirle-----  
Enter the dimension:10  
Area=314.2 units squared  
-----Square-----  
Enter the dimension:5  
Area=25 units squared
```

## **8. Interfaces**

---

Monday, 16th October, 2017

## Problem 8

Write a program to implement Interface and write a program to create an interface Area and implement the same in different classes Rectangle, circle and triangle.

### Area.java

```
public interface Area
{
    final double pi=3.14159;
    public void read_data();
    public void calc_area();
}
```

### Rectangle.java

```
import java.util.*;
public class Rectangle implements Area
{
    double l,b;
    public void read_data()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter length:");
        l=sc.nextDouble();
        System.out.print("Enter breadth:");
        b=sc.nextDouble();
    }

    public void calc_area()
    {
        if(l!=0 && b!=0)
        {
            System.out.print("Area="+l*b+" units squared\n");
        }
        else
        {
            System.out.print("Rectangle doesn't exist\n");
        }
    }
}
```

## Triangle.java

```
import java.util.*;
public class Triangle implements Area
{
    double a,b,c;
    public void read_data()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter length of side 1:");
        a=sc.nextDouble();
        System.out.print("Enter length of side 2:");
        b=sc.nextDouble();
        System.out.print("Enter length of side 3:");
        c=sc.nextDouble();
    }

    public void calc_area()
    {
        if(a+b>c && b+c>a && a+c>b)
        {
            double s=(a+b+c)/2;
            System.out.print("Area="+Math.sqrt(s*(s-a)*(s-b)*(s-c))+" units squared\n");
        }
        else
        {
            System.out.print("Triangle doesn't exist");
        }
    }
}
```

## Circle.java

```
import java.util.*;
public class Circle implements Area
{
    double r;
    public void read_data()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter radius:");
        r=sc.nextDouble();
    }

    public void calc_area()
    {
        if(r!=0)
        {
            System.out.print("Area="+pi*r*r+" units squared\n");
        }
    }
}
```



```
    else
    {
        System.out.print("Circle doesn't exist\n");
    }
}
}
```

### DemoInterface.java

```
import java.util.*;
import java.lang.Math.*;
class DemoInterface
{
    public static void main(String[] args)
    {
        Rectangle R=new Rectangle();
        Circle C=new Circle();
        Triangle T=new Triangle();
        R.read_data();
        R.calc_area();
        C.read_data();
        C.calc_area();
        T.read_data();
        T.calc_area();
    }
}
```

## Output

```
Test Case #1
Enter length:5
Enter breadth:4
Area=20.0 units squared
Enter radius:10
Area=314.159 units squared
Enter length of side 1:6
Enter length of side 2:8
Enter length of side 3:10
Area=24.0 units squared
```

```
Test Case #2
Enter length:0
Enter breadth:4
Rectangle doesn't exist
Enter radius:0
Circle doesn't exist
Enter length of side 1:3
Enter length of side 2:4
Enter length of side 3:7
Triangle doesn't exist
```

## 9. Packages

---

Thursday, 12th October, 2017

# Problem 9

Write a program to implement packages.

## Circle.java

```
package shape;
import java.util.*;
public class Circle
{
    double r;
    public void read_Data()
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter the radius:");
        r=sc.nextFloat();
    }

    public void area_cir()
    {
        // A =  $\pi r^2$ 
        System.out.println("Area is "+3.142*Math.pow(r,2)+" units squared");
    }
}
```

## Rectangle.java

```
package shape;
import java.util.*;
public class Rectangle
{
    double w,l; //sides
    public void read_data()
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter the length:");
        l=sc.nextFloat();
        System.out.print("Enter the breadth:");
        w=sc.nextFloat();
    }
    public void area_rec()
    {
        // A = w * l
        System.out.println("Area is "+ w*l+ " units squared");
    }
}
```

## DemoPackage.java

```
import shape.*;
class DemoPackage
{
    public static void main(String[] args)
    {
        Circle c1=new Circle();
        c1.read_Data();
        c1.area_cir();
        Rectangle r1= new Rectangle();
        r1.read_data();
        r1.area_rec();
    }
}
```

## Output

```
Enter the radius:10
Area is 314.2 units squared
Enter the length:5
Enter the breadth:4
Area is 20.0 units squared
```

# **10. Exception Handling**

---

Thursday, 26th October, 2017

# Problem 10

Write a program to demonstrate try, catch, throw, throws and finally.

```
import java.io.*;
class DemoException
{
    public static void main(String[] args)throws IOException
    {
        try
        {
            InputStreamReader read=new InputStreamReader(System.in);
            BufferedReader in=new BufferedReader(read);
            int n=Integer.parseInt(in.readLine());
        }
        catch(IOException exception)
        {
            System.out.println(exception);
        }
        //int n=Integer.parseInt(in.readLine());
        finally
        {
            System.out.println("Entered Finally");
        }
        int a=5;
        int b=0;
        int c=0;
        int arr[]={1,2};
        try
        {
            c=a/b;
            System.out.println("exception" +c);
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
            throw new ArrayIndexOutOfBoundsException(arr[3]);
        }
    }
}
```

## Output

```
635
Entered Finally
java.lang.ArithmeticException: / by zero
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
    at DemoException.main(DemoException.java:33)
```

# **11. Working of an Array**

---

Monday, 16th October, 2017



# Problem 11.1

Write a program to implement Binary Search.

```
import java.util.*;
class ArraySearch
{
    public static void main(String[] args)
    {
        int A[] = {1,1,5,6,7,8,9,10,25}; //array purposefully initialized in sorted manner
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the element you want to search for:");
        int x = sc.nextInt();
        if(bsearch(A,0,A.length-1,x)==true)
        {
            System.out.print(x+" is present in the array");
        }
        else
        {
            System.out.print(x+" is not present in the array");
        }
    }

    static boolean bsearch(int A[],int l,int r,int x)
    {
        if(l>r)
        {
            return false;
        }
        else
        {
            int m=(l+r)/2;
            if(A[m]==x)
            {
                return true;
            }
            else if(A[m]>x)
            {
                return bsearch(A,0,m-1,x);
            }
            else
            {
                return bsearch(A,m+1,r,x);
            }
        }
    }
}
```

## Output

```
Test Case #1
Enter the element you want to search for:8
8 is present in the array
```

```
Test Case #2
Enter the element you want to search for:12
12 is not present in the array
```

## Problem 11.2

Write a program to implement Bubble Sort.

```
import java.util.*;
class ArraySort
{
    public static void main(String[] args)
    {
        int[] A= new int[100];
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter no. of elements in the array:");
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
        {
            System.out.print("Enter an element:");
            A[i]=sc.nextInt();
        }
        //Bubble sort starts
        for(int i=0;i<n;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(A[i]>A[j])
                {
                    int temp=A[i];
                    A[i]=A[j];
                    A[j]=temp;
                }
            }
        }
        //Printing the sorted array
        System.out.println("The sorted array is....");
        for(int i=0;i<n;i++)
        {
            System.out.print(A[i]+" ");
        }
    }
}
```

## Output

```
Enter no. of elements in the array:10
Enter an element:3
Enter an element:7
Enter an element:1
Enter an element:9
Enter an element:0
Enter an element:1
Enter an element:3
Enter an element:8
Enter an element:9
Enter an element:03
The sorted array is....
0 1 1 3 3 3 7 8 9 9
```

## Problem 11.3

Write a program to implement matrix multiplication.

```
import java.util.*;
import java.lang.*;
class MatMultiply
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        double[][] A=new double[100][100],B=new double[100][100],C=new double[200][200];
        System.out.print("Enter no.of rows of first matrix:");
        int p=sc.nextInt();
        System.out.print("Enter no.of columns of first matrix:");
        int q=sc.nextInt();
        System.out.print("Enter no.of rows of second matrix:");
        int r=sc.nextInt();
        System.out.print("Enter no.of columns of second matrix:");
        int s=sc.nextInt();
        if(q!=r)
        {
            System.out.print("The matrices are multiplication incompatible");
        }
        //Filling up matrices
        else
        {
            System.out.println("For matrix A.....");
            read_array(A,p,q);
            System.out.println("A=");
            disp_array(A,p,q);
            System.out.println("For matrix B.....");
            read_array(B,r,s);
        }
    }
}
```

```

        System.out.println("B=");
        disp_array(B,r,s);
        C=mat_multiply(A,B,p,q,s);
        System.out.println("A X B=");
        disp_array(C,p,s);
    }
}

static void read_array(double A[][],int m,int n)
{
    Scanner sc=new Scanner(System.in);
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            System.out.print("Enter an element:");
            A[i][j]=sc.nextDouble();
        }
    }
}

static void disp_array(double A[][],int m,int n)
{
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            System.out.print(A[i][j]+" ");
        }
        System.out.println();
    }
}

static double[][] mat_multiply(double A[][],double B[][],int m,int p,int n)
{
    double[][] C=new double[200][200];
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            C[i][j]=0.0;
            for(int k=0;k<p;k++)
            {
                C[i][j]+=(A[i][k]*B[k][j]);
            }
        }
    }
    return C;
}
}

```

## Output

```
Test Case #1
Enter no.of rows of first matrix:3
Enter no.of columns of first matrix:3
Enter no.of rows of second matrix:4
Enter no.of columns of second matrix:4
The matrices are multiplication incompatible

Test Case #2 For the sake of demonstration matrices which are inverses of
each other have been selected for this test case
Enter no.of rows of first matrix:3
Enter no.of columns of first matrix:3
Enter no.of rows of second matrix:3
Enter no.of columns of second matrix:3
For matrix A.....
Enter an element:1
Enter an element:-1
Enter an element:1
Enter an element:2
Enter an element:-1
Enter an element:0
Enter an element:1
Enter an element:0
Enter an element:0
A=
1.0 -1.0 1.0
2.0 -1.0 0.0
1.0 0.0 0.0
For matrix B.....
Enter an element:0
Enter an element:0
Enter an element:1
Enter an element:0
Enter an element:-1
Enter an element:2
Enter an element:1
Enter an element:-1
Enter an element:1
B=
0.0 0.0 1.0
0.0 -1.0 2.0
1.0 -1.0 1.0
A X B=
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
```

## **12. Working of String**

---

Monday, 16th October, 2017

# Problem 12.1

Write a program to reverse a string and decide whether it is palindrome or not.

```
import java.util.*;
class Palindrome
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter a string:");
        String S=sc.nextLine();
        boolean f=true;
        for(int i=0;i<S.length()/2;i++)
        {
            if(S.charAt(i)!=S.charAt(S.length()-i-1))
            {
                f=false;
                break;
            }
        }
        if(f)
        {
            System.out.print(S+" is a palindrome");
        }
        else
        {
            System.out.print(S+" is not a palindrome");
        }
    }
}
```

## Output

Test Case #1

Enter a string:abcdcba

abcdcba is a palindrome

Test Case #2

Enter a string:abbcba

abbcba is not a palindrome

## Problem 12.2

Write a program to capitalize the first letter of each word

```
import java.util.*;
class FirstUpper
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String S=new String();
        System.out.print("Enter a string:");
        S=sc.nextLine();
        String NewS="";
        boolean sp=false;
        for(int i=0;i<S.length();i++)
        {
            if(i==0)
            {
                NewS+=Character.toUpperCase(S.charAt(0));
            }
            else if(S.charAt(i)==' ')
            {
                sp=true;
                NewS+=' ';
            }
            else
            {
                if(sp==true)
                {
                    NewS+=Character.toUpperCase(S.charAt(i));
                    sp=false;
                }
                else
                {
                    NewS+=S.charAt(i);
                }
            }
        }
        System.out.print("New String- "+NewS);
    }
}
```



## Output

```
Test Case #1
Enter a string:my name is pronoy mandal
New String- My Name Is Pronoy Mandal
```

```
Test Case #2
Enter a string:fhgsfhg hsdghsgd
New String- Fhgsfhg Hsdghsgd
```

## Problem 12.3

Write a program to Count the frequency of particular letter and replace that particular letter by #

```
import java.util.*;
class Frequency
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter a string:");
        String S=sc.nextLine();
        System.out.print("Enter the charcter whose frequency you want to find out:");
        char x=sc.nextLine().charAt(0);
        int f=0;
        for(int i=0;i<S.length();i++)
        {
            if(S.charAt(i)==x)
            {
                f++;
            }
        }
        System.out.println("Frequency of "+x+" in the given string: "+f);
        S=S.replace(x,'#');
        System.out.print("New String: "+S);
    }
}
```

## Output

```
Test Case #1
Enter a string:aaab ccd beeart
Enter the charcter whose frequency you want to find out:a
Frequency of a in the given string: 4
New String: ###b ccd bee#rt
```

```
Test Case #2
Enter a string:abcdefg
Enter the charcter whose frequency you want to find out:h
Frequency of h in the given string: 0
New String: abcdefg
```

## Problem 12.4

Write a program to count the no. of uppercase, lowercase, blank spaces, digits, special characters from string

```
import java.util.*;
class StringStat
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter a string:");
        String S=sc.nextLine();
        int u=0,l=0,d=0,s=0,spch=0;
        for(int i=0;i<S.length();i++)
        {
            if(S.charAt(i)==' ')
            {
                s++;
            }
            else if(S.charAt(i)>='A' && S.charAt(i)<='Z')
            {
                u++;
            }
            else if(S.charAt(i)>='a' && S.charAt(i)<='z')
            {
                l++;
            }
            else if(S.charAt(i)>='0' && S.charAt(i)<='9')
            {
                d++;
            }
        }
    }
}
```

```
        else
        {
            spch++;
        }
    }
    System.out.println("-----String Statistics-----");
    System.out.print("No. of Uppercase letters- "+u+"\nNo. of Lowercase letters- "+l+"\nNo. of Whitespaces- "+s+"\nNo. of digits- "+d+"\nNo. of Special characters- "+spch);
}
}
```

## Output

```
Enter a string:fhfdh FGE#$@ sfgf1905
-----String Statistics-----
No. of Uppercase letters- 3
No. of Lowercase letters- 9
No. of Whitespaces- 2
No. of digits- 4
No. of Special characters- 3
```

# **13. Multithreading**

---

Thursday, 26th October, 2017

# Problem 13

Write a program to demonstrate multithreading.

```
class MultiThread extends Thread
{
    public void run()
    {
        System.out.println("Running Thread Name: "+ this.currentThread().getName());
        System.out.println("Running Thread Priority: "+ this.currentThread().getPriority());
    }
}

public class MultiThreading
{
    public static void main(String[] args)
    {
        MultiThread multiThread1 = new MultiThread();
        multiThread1.setName("Thread1");
        multiThread1.setPriority(Thread.MIN_PRIORITY);
        MultiThread multiThread2 = new MultiThread();
        multiThread2.setName("Thread2");
        multiThread2.setPriority(Thread.MAX_PRIORITY);
        MultiThread multiThread3 = new MultiThread();
        multiThread3.setName("Thread3");
        multiThread1.start();
        multiThread2.start();
        multiThread3.start();
    }
}
```

## Output

```
Running Thread Name: Thread3
Running Thread Priority: 5
Running Thread Name: Thread1
Running Thread Priority: 1
Running Thread Name: Thread2
Running Thread Priority: 10
```

## **14. Applets**

---

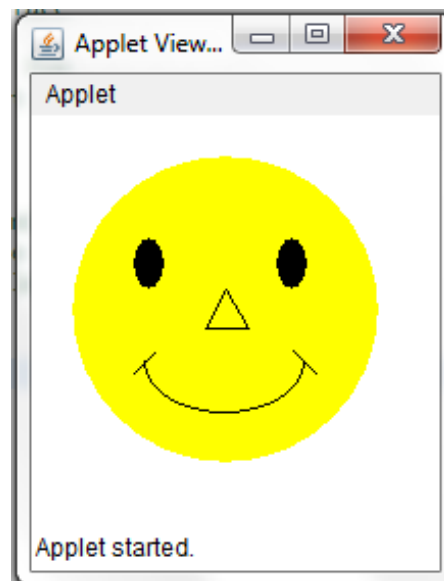
Thursday, 26th October, 2017

# Problem 14

Write a program on Applet to demonstrate Graphics, Font and Color class

```
import java.awt.*;
import java.applet.*;
public class Smiley extends Applet
{
    public void paint(Graphics g)
    {
        Font f = new Font("Helvetica", Font.BOLD,20);
        g.setFont(f);
        g.setColor(Color.yellow);
        g.fillOval(20,20,150,150);    // For face
        g.setColor(Color.black);
        g.fillOval(50,60,15,25);      // Left Eye
        g.fillOval(120,60,15,25);     // Right Eye
        int x[] = {95,85,106,95};
        int y[] = {85,104,104,85};
        g.drawPolygon(x,y,4);          // Nose
        g.drawArc(55,95,78,50,0,-180); // Smile
        g.drawLine(50,126,60,116);    // Smile arc1
        g.drawLine(128,115,139,126);  // Smile arc2
    }
}
```

## Output



# 15. GUI

---

Thursday, 26th October, 2017



# Problem 15

Write a program to create GUI application with event handling using AWT controls.

```
public class DemoEvents extends java.applet.Applet
{
    public void init()
    {
        try
        {
            java.awt.EventQueue.invokeLater(new Runnable()
            {
                public void run()
                {
                    initComponents();
                }
            });
        }
        catch (Exception ex)
        {
            ex.printStackTrace();
        }
    }
    /**
     * This method is called from within the init() method to initialize the
     * form. WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        N1 = new java.awt.Label();
        tN1 = new java.awt.TextField();
        N2 = new java.awt.Label();
        tN2 = new java.awt.TextField();
        tN3 = new java.awt.TextField();
        N3 = new java.awt.Label();
        AVG = new java.awt.Label();
        tAVG = new java.awt.TextField();
        B = new java.awt.Button();

        N1.setName(""); // NOI18N
        N1.setText("Number 1");

        N2.setText("Number 2");

        N3.setText("Number 3");

        AVG.setText("Average");
```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(tN1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(N1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(19, 19, 19)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(N2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(tN2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(22, 22, 22)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(tN3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(N3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(AVG, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(tAVG, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 37,
Short.MAX_VALUE)
        .addComponent(B, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(28, Short.MAX_VALUE))
    );
} // </editor-fold>

private void BActionPerformed(java.awt.event.ActionEvent evt) {
    double n1=Double.parseDouble(tN1.getText());
    double n2=Double.parseDouble(tN2.getText());
    double n3=Double.parseDouble(tN3.getText());
    tAVG.setText(" "+(n1+n2+n3)/3);

}
// Variables declaration - do not modify
private java.awt.Label AVG;
private java.awt.Button B;
private java.awt.Label N1;
private java.awt.Label N2;
private java.awt.Label N3;
private java.awt.TextField tAVG;
private java.awt.TextField tN1;
private java.awt.TextField tN2;
private java.awt.TextField tN3;
// End of variables declaration
}

```

# Output

