83 - JavaFX

66 – Generic method

64 - Collections

Java Programming
CSE 1007

Lab Assignment 1
Arrays and Loops

Submitted by
Om Ashish Mishra
16BCE0789

Question 1
Write a Java program to display all the prime numbers within a range.

Code

```
import java.util.*;
class Question1
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
   int a, b, c;      int i, j;
                 System.out.println("Enter the
range");                    a=sc.nextInt();
 b=sc.nextInt();

                for(i=a;i<=b;i++)
                {
                        c=0;
                        for(j=2;j<i/2;j++)
                        {
                                if(i%j==0)
                                  {
                                  c=1;
                          break;

        }
                        }
```

```
                        if(c==0)
                        {
                                System.out.print(i+" ");
                        }

                }
        }
}
```

Output

Question 2

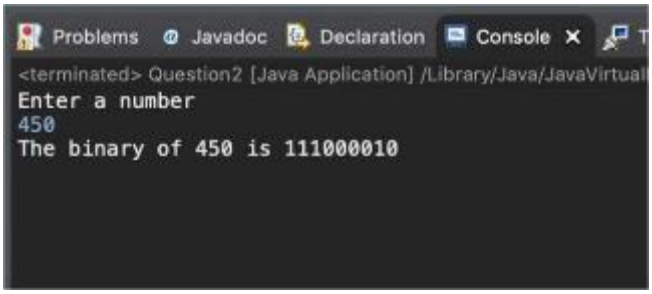
Write a Java program to convert a decimal number to its equivalent binary number. Eg: $25_{10} = 11001_2$

Code

```
import java.util.*;
public class Question2
{
        public static void main(String args[])
        {
                //Program to convert a decimal number into its equivalent binary number
                  Scanner sc = new Scanner(System.in);
                  int num;
                 System.out.println("Enter a number");
                  num =
sc.nextInt();               String
bin="";                     int rem, temp=num;
```

```
                while(temp>0)
                {
                        rem = temp%2;
                        bin = Integer.toString(rem)+bin;
                        temp/=2;


                }
                System.out.println("The binary of "+num+" is "+bin);
        }
}
```

Output



Question 3

Write a Java program to print the following patterns by reading the number of lines from the user.

1.

```
*
**
***
**** *****
```

2.

```
      *
* *
* * *
* * * *
* * * * *
```

3.

```
     *
* *
* * *
* * * *
* * *
* *
     *
```

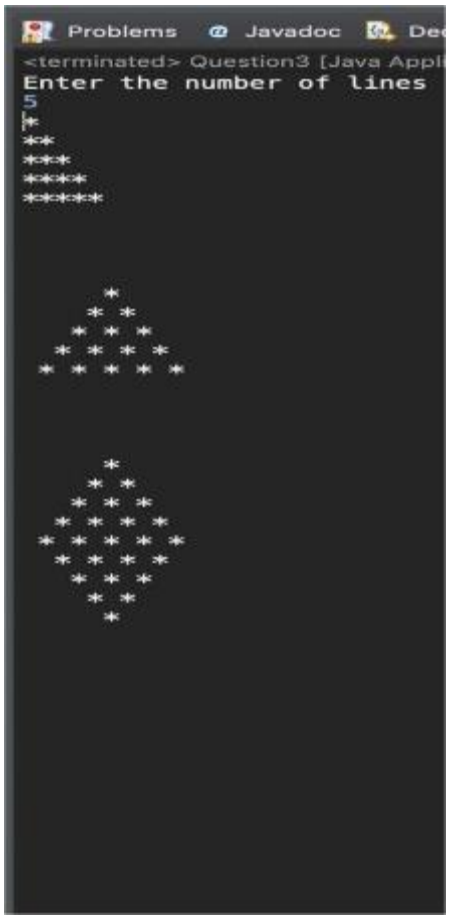
Code

```
import java.util.*;
public class Question3
{
        public static void main(String args[])
        {
                //Printing Patterns
                Scanner sc = new Scanner(System.in);
                int n,c;
                 System.out.println("Enter the number of
lines");                  n = sc.nextInt();                  int
i,j,k;                //Printing pattern1


                for(i=1;i<=n;i++)
                {
                        for(j=1;j<=i;j++)
                        {
                                System.out.print("*");
                        }
                        System.out.println();
                }
                //Printing Pattern 2
 System.out.println("\n\n\n");
```

```
        c=n;
        for(i=1;i<=n;i++)
        {
                for(k=1;k<=c;k++)
                {
                        System.out.print(" ");
                }
                c-=1;
                for(j=1;j<=i;j++)
                {
                        System.out.print("* ");
                }
                System.out.println();
        }
        System.out.println("\n\n\n");

         //Printing Pattern 3
c=n;
        for(i=1;i<=n;i++)
        {
                for(k=1;k<=c;k++)
                {
                        System.out.print(" ");
                }
                c-=1;
                for(j=1;j<=i;j++)
                {
                        System.out.print("* ");
                }
                System.out.println();
        }

        c=2;

        for(i=n-1;i>=1;i--)
        {

                for(k=1;k<=c;k++)
                {
                        System.out.print(" ");
                }
                c+=1;
                for(j=1;j<=i;j++)
                {
                         System.out.print("* ");
```

```
                    }
                    System.out.println();
                }

        }
}
```

Output



Question 4

Write a Java program to sum up all the digits of an integer till the sum is a single digit. Eg: INPUT = 9985 9+9+8+5 = 31

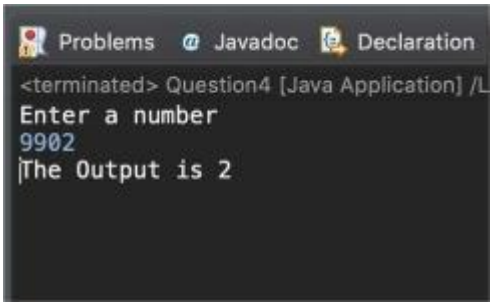
3+1 = 4 OUTPUT = 4

Code

```
import java.util.*;

public class Question4 {  public static void
main(String args[])
        {
```

```
                Scanner sc = new Scanner(System.in);
                int n;
               System.out.println("Enter a number");
               n = sc.nextInt();

  //Finding the sum of digits till the number is a single digit     int num=n;
               int digit, sum=num;
                do
                {

       num = sum;
                        sum=0;
                        while(num>0)
                        {
                                digit = num%10;
                         sum+=digit;
                                num/=10;
                        }
               }
               while(sum>=10);

               System.out.println("The Output is "+sum);
       }
}
```

Output



Question 5

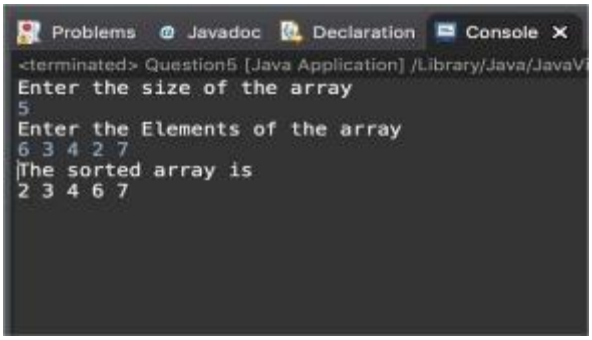
Write a Java program to sort a numerical array using selection sort algorithm and remove all the duplicates from the same array. [Hint: Use single array]

Code

```
import java.util.*;
public class Question5
{
	public static void main(String args[])
	{
		//Selection sort
		 Scanner sc = new Scanner(System.in);
		 int len;
		int i, j, temp;
  System.out.println("Enter the size of the array");   len =
sc.nextInt();
		System.out.println("Enter the Elements of the array");
		int a[] = new int[10];
		for(i=0;i<len;i++)
		{
			a[i] = sc.nextInt();
		}

		 //Selection sort algorithm
	 int minpos = 0;
		for(i=0;i<len-1;i++)
		{
			minpos=i;
			for(j=i+1;j<len;j++)
			{
				if(a[j]<a[minpos])
					minpos = j;
			}
			//Swapping
  temp = a[minpos];    a[minpos] = a[i];
			a[i] = temp;
		}
		System.out.println("The sorted array is ");
		for(i=0; i<len;i++)
		{
			System.out.print(a[i]+" ");
		}
		System.out.println();
	}
```

Question 6
}

Output

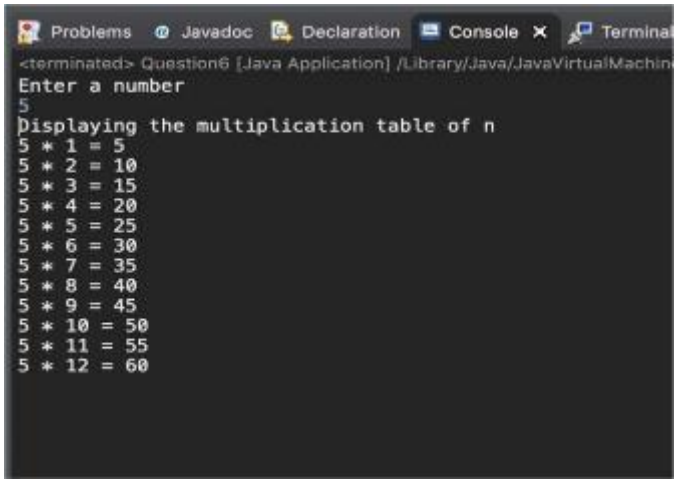


Question 6
Write a Java program to read an integer ‘n’ from the user and display the multiplication table of ‘n’.

Code

```
import java.util.*; public class Question6 {
public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                 int n;
                System.out.println("Enter a number");
                n = sc.nextInt();
   System.out.println("Displaying the multiplication table of n");    for(int
i=1;i<=12;i++)
                {
                        System.out.println(n+" * "+i+" = "+(n*i));
                }
        }
}
```

Question 7



Output

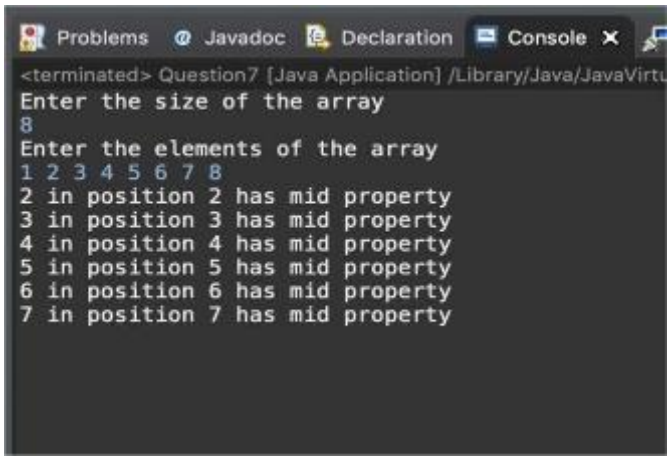
Write a Java program to list out the elements in an array having mid property. An element in an array is said to have the mid property if its left element is lesser than it and also the right element is greater than it. Eg: ..... , 3, 5, 9, ...

5 is having mid property.

Code

```
import java.util.*;
public class Question7
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		int len;
		//System.out.println();
		System.out.println("Enter the size of the array");
   len = sc.nextInt();	int a[] = new
int[len];	int i;
		System.out.println("Enter the elements of the array");
		for(i=0;i<len;i++)
		{
			a[i] = sc.nextInt();
		}
		for(i=1;i<len-1;i++)
		{
			if(a[i-1]<a[i] && a[i+1]>a[i])
			{
				System.out.println(a[i]+" in position "+(i+1)+" has mid property");
			}
		}

	}
}
```

Question 8



Output

Question 9

Print Hailstone sequence for a number.

(Note: Take any positive integer n. If n is even, divide it by 2 to get n / 2. If n is odd, multiply it by 3 and add 1 to obtain 3n + 1. Repeat the process indefinitely. The conjecture is that no matter what number you start with, you will always eventually reach 1.)

Eg. Hailstone sequence of 15 is

15, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1

Code

```
import java.util.*;
public class Question8
{
        public static void main(String args[])
        {
   Scanner sc = new Scanner(System.in);     int n;
   System.out.println("Enter a number");     n =
sc.nextInt();
   //Printing the hailstone sequence      int num = n;
                while(num!=1)
                {

                        if(num%2 == 0)
                          {
                          num/=2;
                          }
                 else
                        {
                                num = 3*num +1;
                        }
                        System.out.print(num+" ");
                }

        }
}
```

Output

Question 10

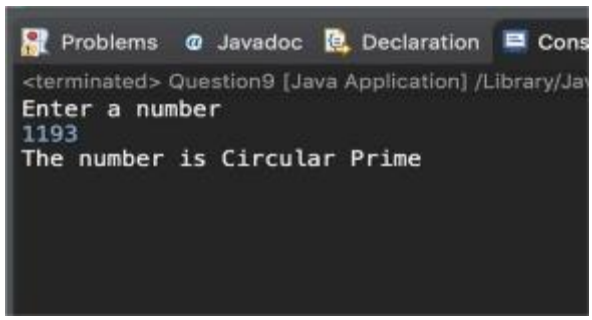
Find whether an entered number is CIRCULAR PRIME or not. Display YES if it is a circular prime, otherwise display NO. A circular prime number is a number that remains prime on any cyclic rotation of its digits (in base 10).
For example 1193 is circular prime because 1931, 9311, 3119 and 1193 are all prime numbers. Code

```
import java.util.*;
public class Question9
{
        static boolean isprime(int num)
        {
                int i, c=0;
                for(i=2;i<num/2;i++)
                {
                        if(num%i==0)
                          {
                          c+=1;
                 break;
                        }
                 }
                 if(c==0)
                 return true;
          else
                        return false;


        }
        static String permute(String s)
        {

                return s.substring(1)+ s.substring(0,1);
        }

        public static void main(String args[])
        {
 Scanner sc = new Scanner(System.in);   int n;
```

Question 11

```
   System.out.println("Enter a number");    n =
sc.nextInt();
   //Step 1: Permuting the numbers    int i, num,
c=0;

                String s = Integer.toString(n);
                for(i=0;i<s.length();i++)
                {
                        num = Integer.parseInt(s);
                        if(!isprime(num))
                          {
                          c+=1;


                                break;
                        }
                          //Permute
                          s =
permute(s);
                }
                if(c==0)
                        System.out.println("The number is Circular Prime");
                else
                        System.out.println("The number is not Circular Prime");
        }
}
```

Output



Question 10

Write a Java program to find out the greatest common divisor of two input values using a function. Code

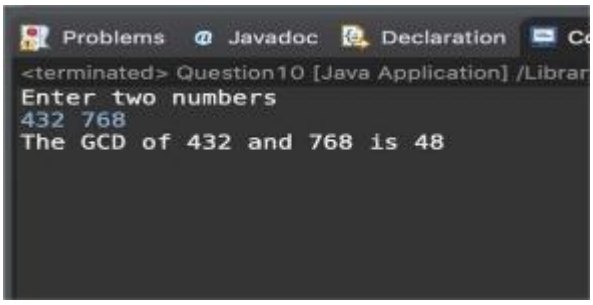
```
import java.util.*;
public class Question10
{
```

Question 12

```
 //Function to return the GCD of two numbers    static int
gcd(int a, int b)
        {
                 int rem=1;
 while(rem!=0)
                {
                        rem = b%a;
                 b=a;                a=rem;

                }
                return b;

        }
        public static void main(String args[])
        {
                 Scanner sc = new Scanner(System.in);
                 int a, b;
   System.out.println("Enter   two   numbers");      a   =
sc.nextInt();  b = sc.nextInt();
```

```
                System.out.println("The GCD of "+a+" and "+b+" is "+ gcd(a,b));
        }
}
```

Output



Question 11

Write a Java program to reverse the contents of the array using different functions for different types of array (without using any secondary array for reversing).

Code

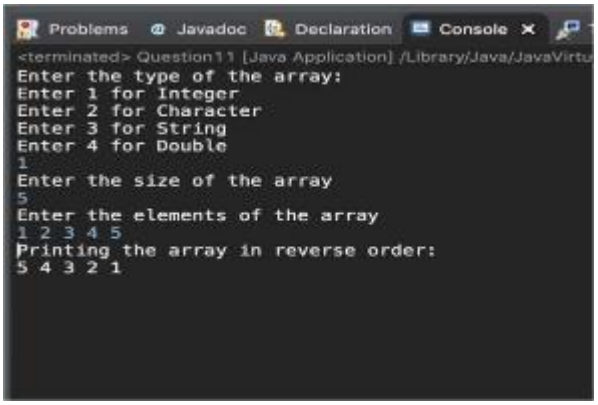
```
import java.util.*;

public class Question11
{
        static int[] revint(int a[])
        {
                int len = a.length;
                int end = len-1, temp;

                for(int i=0;i<len/2;i++)
                {
                        temp = a[i];
         a[i]=a[end];
 a[end]=temp;
                        end-=1;
                }

                return a;
        }

        static char[] revchar(char a[])
        {
  int len = a.length;   int end =
len-1;   char temp;

                for(int i=0;i<len/2;i++)
```

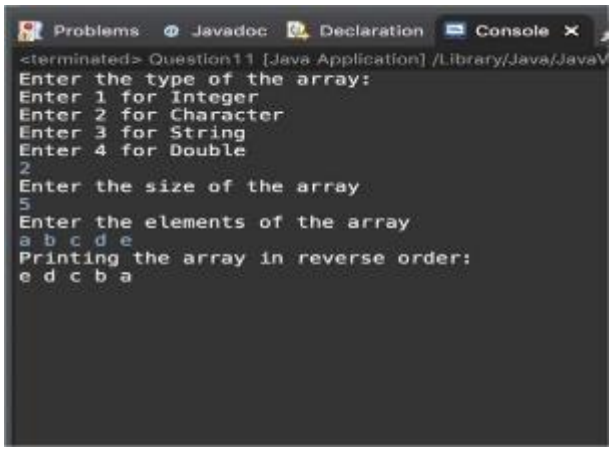
```
                {
                        temp = a[i];
        a[i]=a[end];
a[end]=temp;
                 end-=1;
        }

                return a;
        }

        static String[] revstring(String a[])
        {
                 int len = a.length;
        int end = len-1;
String temp;

                for(int i=0;i<len/2;i++)
                {
                        temp = a[i];
        a[i]=a[end];
a[end]=temp;
                        end-=1;
                }


                return a;
        }

        static double[] revdouble(double a[])
        {
                 int len = a.length;
 int end = len-1;     double temp;

                for(int i=0;i<len/2;i++)
                {
                        temp = a[i];
        a[i]=a[end];
a[end]=temp;
                        end-=1;
                }

                return a;
        }

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

```
Scanner sc = new Scanner(System.in);
 System.out.println("Enter the type of the array:\nEnter 1 for Integer \nEnter 2 for
Character \nEnter 3 for String \nEnter 4 for Double");
int ch=sc.nextInt();
System.out.println("Enter the size of the array");
int len = sc.nextInt();

int i;



System.out.println("Enter the elements of the array");
switch(ch)
{
        case 1:
                int a[] = new int[len];
for(i=0;i<len;i++)
                        a[i] = sc.nextInt();
                System.out.println("Printing the array in reverse order:
"); a    = revint(a);                    for(i=0;i<len;i++)

System.out.print(a[i]+" ");
                break;
case 2:
                char b[] = new char[len];
for(i=0;i<len;i++)
                        b[i] = sc.next().charAt(0);
                System.out.println("Printing the array in reverse order:
"); b    = revchar(b);                   for(i=0;i<len;i++)

System.out.print(b[i]+" ");
                break;
case 3:
                sc.nextLine();
                String c[] = new
String[len];                            for(i=0;i<len;i++)
                c[i] = sc.nextLine();
 for(i=0;i<len;i++)
                        System.out.print(c[i]+" ");
                    System.out.println();
                System.out.println("Printing the array in reverse order:
"); c    = revstring(c);                 for(i=0;i<len;i++)

System.out.print(c[i]+" ");
```

```
                                    break;
            case 4:
                                    double d[] = new double[len];
            for(i=0;i<len;i++)
                                        d[i] = sc.nextDouble();
                                    System.out.println("Printing the array in reverse order:
");                                  d = revdouble(d);
 for(i=0;i<len;i++)                                          System.out.print(d[i]+" ");
                            break;                      }



            }
}
```

Output



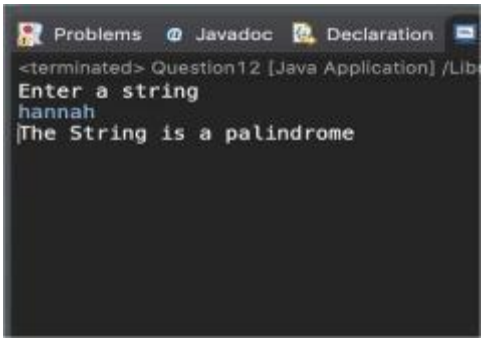


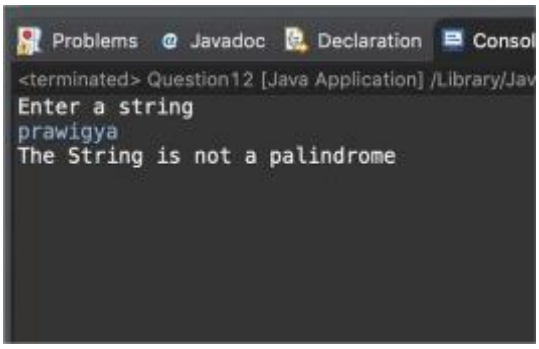
Write a Java program to check the given string is palindrome or not.

Code

```
import java.util.*;
public class Question12
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a string");
		String s= sc.nextLine();
		 String s2="";
	 //Reverse the string
		for(int i=0;i<s.length();i++)
		{
			s2=s.charAt(i)+s2;
		}
		if(s.compareTo(s2)==0)
			System.out.println("The String is a palindrome");
		else
			System.out.println("The String is not a palindrome");

	}

}
```

Output



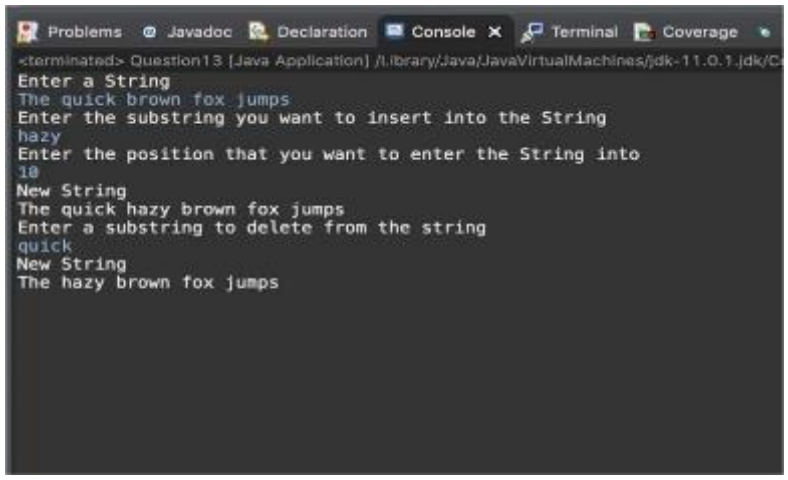


Write a Java program to insert a string into another string and delete a substring from a string.

Code

```
import java.util.*;
public class Question13
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a String");
		String s = sc.nextLine();
		System.out.println("Enter the substring you want to insert into the String");
		String s1 = sc.nextLine();
   System.out.println("Enter the position that you want to enter the String into");     int pos = sc.nextInt();
		 //Enter the substring into the string		String sub1="", sub2="";
		 sub1 = s.substring(0,pos);
	 sub2=s.substring(pos);			s = sub1 + s1 + sub2;
		System.out.println("New String \n"+s);
		//Part 2
		sc.nextLine();
		 System.out.println("Enter a substring to delete from the string");		String s2= sc.nextLine();		int pos2 = s.indexOf(s2);
		s = s.substring(0,pos2) + s.substring(pos2+s2.length());
		System.out.println("New String \n"+s);

	}
}
```

Output

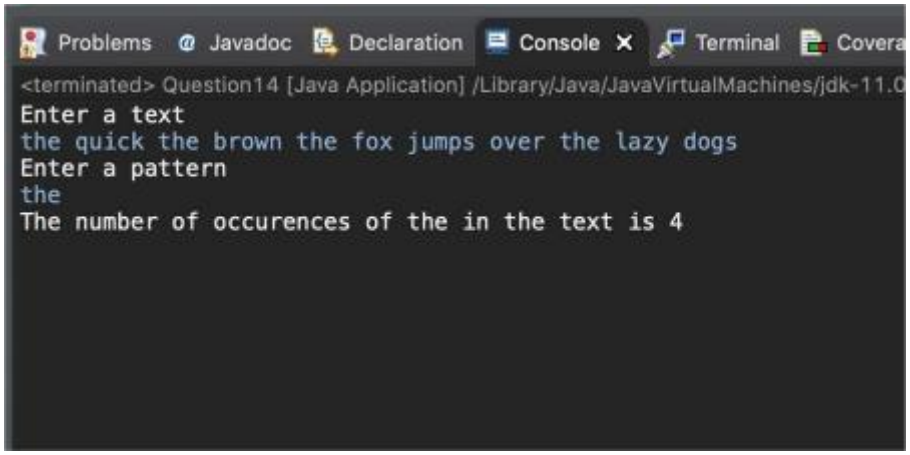


Write a Java program to find out the number of occurrences of a pattern string in a given text.

Code

```
import java.util.*;
public class Question14
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a text");
		String s = sc.nextLine();
		//sc.nextLine();
		 System.out.println("Enter a pattern
");		 String p = sc.nextLine();
	 int c=0,pos=0;

		while(true)
		{
			pos = s.indexOf(p,pos);
			  if(pos==-1)
			  break;
		 c+=1;
			pos+=1;

		}
		System.out.println("The number of occurences of "+p+" in the text is "+c);
	}
}
```

Output



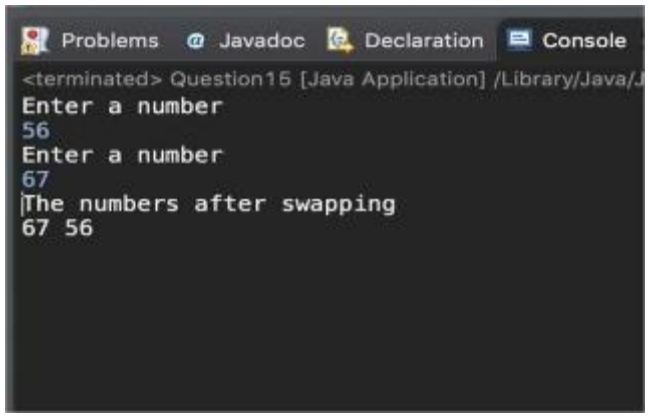
Write a Java program to swap two values in a SWAP( ) method using wrapper classes.

Code

```
import java.util.*; public
class Question15 {
        int a;
        Question15()
        {
a=0;
        }
       //swap 2 values in a method using wrapper class
       static void swap(Question15 ob1, Question15 ob2)
        {
                int temp = ob1.a;
        ob1.a = ob2.a;
                ob2.a = temp;
        }
       public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
   System.out.println("Enter a number");
Question15 ob1 = new Question15(); ob1.a =
sc.nextInt();
   System.out.println("Enter a number");
Question15 ob2 = new Question15(); ob2.a =
sc.nextInt();  swap(ob1, ob2);
               System.out.println("The numbers after swapping");
               System.out.println(ob1.a+" "+ob2.a);
```

```
        }
}
```

Output



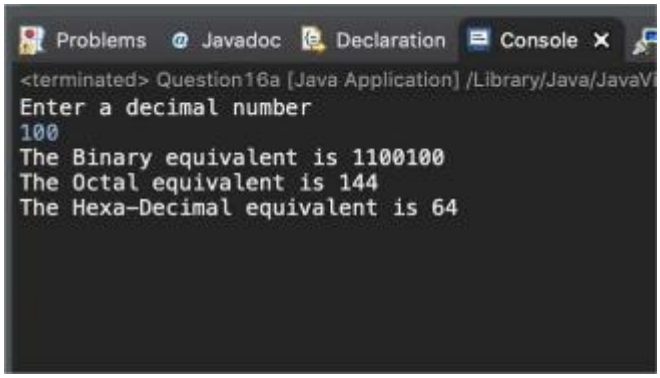
Write a Java program to convert the decimal number to binary, octal, and hexadecimal numbers using wrapper class methods. [Hint: Integer and Long classes]

Code

```
import java.util.*;
public class
Question16a
{   int dec, oct, hex;  long
bin;
        Question16a()
        {
        dec=0;
        oct=0;
 hex=0;
                bin=0;
        }
        static long decToBinary(int d)
        {
                long b = Long.parseLong(Integer.toBinaryString(d));
                return b;
        }

        static int decToOct(int d)
        {
                int a = Integer.parseInt(Integer.toOctalString(d));
                return a;
        }
```

```
        static int decToHex(int d)
        {
                int a = Integer.parseInt(Integer.toHexString(d));
                return a;
        }
        public static void main(String args[])
        {
   Scanner sc = new Scanner(System.in);
System.out.println("Enter a decimal number");
Question16a ob = new Question16a();      ob.dec =
sc.nextInt();    ob.oct = decToOct(ob.dec);    ob.hex
=
decToHex(ob.dec);                ob.bin = decToBinary(ob.dec);
                System.out.println("The Binary equivalent is "+ob.bin);
                System.out.println("The Octal equivalent is "+ob.oct);
                System.out.println("The Hexa-Decimal equivalent is "+ob.hex);
        }

}
```

Output



Question 17

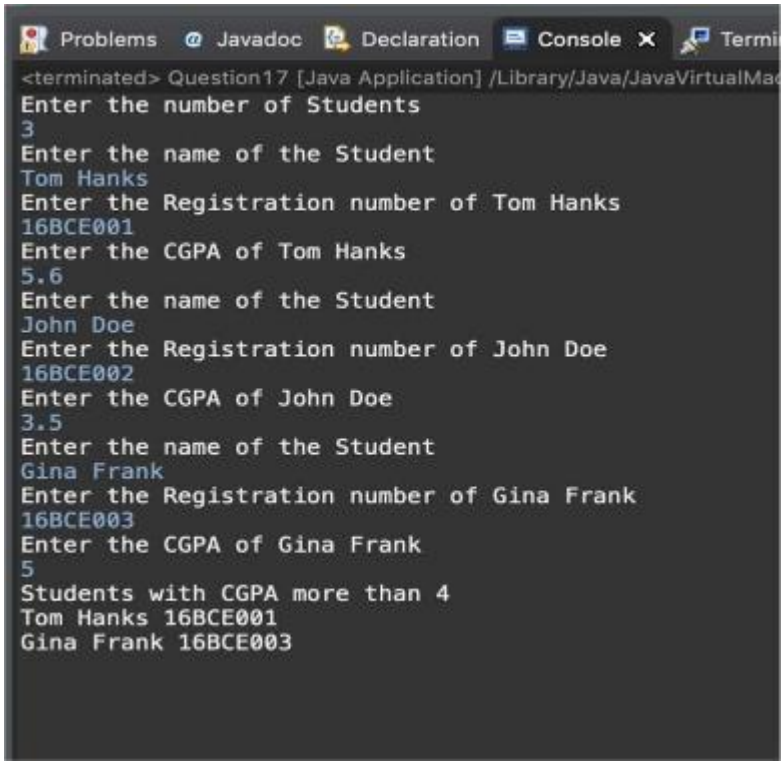
Write a class definition for ‘stu’ with name, regno, and cgpa values and required methods as members of the class. Create an array of objects of ‘stu’ for ‘n’ number of students in G2 slot. Write a Java program to display the name and registration numbers of the students who have CGPA less than 4 in G2 slot.

Code

```
import java.util.*;
public class Question17
{
        String name, regno;
double cgpa;
        Question17()
        {
        name = "";
regno="";
                cgpa = 0.0;
        }
        void init(Question17 ob)
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the name of the Student");
                ob.name= sc.nextLine();
                System.out.println("Enter the Registration number of "+ob.name);
                ob.regno = sc.nextLine();
                System.out.println("Enter the CGPA of "+ob.name);
                ob.cgpa = sc.nextDouble();

        }
```

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

		int n;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the number of Students");
		n = sc.nextInt();

		Question17[] StudentArray = new Question17[n];
		int i;
		for(i = 0;i<n;i++)
		{
			StudentArray[i] = new Question17();
			StudentArray[i].init(StudentArray[i]);
		}

		System.out.println("Students with CGPA more than 4");
		for(i = 0;i<n;i++)
		{
			if(StudentArray[i].cgpa>4)
				System.out.println(StudentArray[i].name+" "+StudentArray[i].regno);
		}
	}
}
```

Output



Question 18

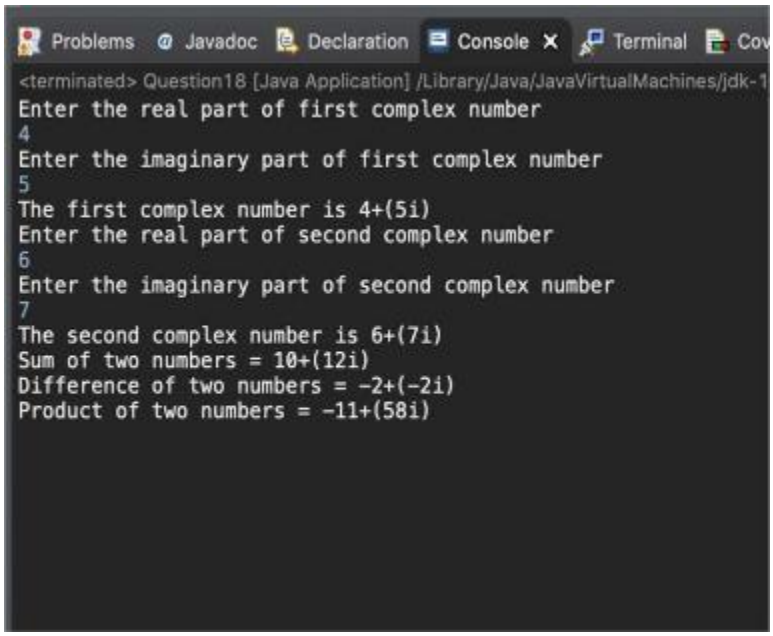
Write a Java program to implement complex number arithmetic using classes and use multiple constructors for initialising the complex numbers.

Code

```
import java.util.*;
public class Question18
{   int
real;  int
imag;
        Question18()
        {
                real = 0;
                 imag =
0;       }

        Question18(int real, int imag)
        {
   this.real = real;      this.imag =
imag;
        }
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                Question18 ob1 = new Question18();
  Question18 ob2 = new Question18();   int r, i;
  System.out.println("Enter the real part of first complex number");   r =
sc.nextInt();
  System.out.println("Enter the imaginary part of first complex number");   i =
sc.nextInt();
                 ob1 = new Question18(r,i);
                System.out.println("The first complex number is "+ob1.real+"+("+ob1.imag+"i)");

  System.out.println("Enter the real part of second complex number");   r =
sc.nextInt();
                  System.out.println("Enter the imaginary part of second complex
number");                 i = sc.nextInt();                 ob2 = new Question18(r,i);
                System.out.println("The second complex number is
"+ob2.real+"+("+ob2.imag+"i)");       int
tempr, tempi;    //Adding the two numbers
tempr = ob1.real +
ob2.real;                 tempi = ob1.imag +
ob2.imag;
                System.out.println("Sum of two numbers = "+tempr+"+("+tempi+"i)");
   //Subtracting the two numbers     tempr =
ob1.real - ob2.real;      tempi = ob1.imag -
ob2.imag;
```

```
                System.out.println("Difference of two numbers = "+tempr+"+("+tempi+"i)");
  //Multiplying the two numbers   tempr = ob1.real*ob2.real
                 - ob1.imag*ob2.imag;
                 tempi = ob1.imag*ob2.real + ob1.real*ob2.imag;
                System.out.println("Product of two numbers = "+tempr+"+("+tempi+"i)");



        }

}
```

Output



Question 19

Write a Java program to print a pattern using a method PRINT( ) as follows: *

**

***

****

*****

The type of the character and/or the number of lines to be printed can be taken as input from the user. The default values are '*' and 5. Using the concept of method overloading write different definitions for PRINT( ) with different argument list.

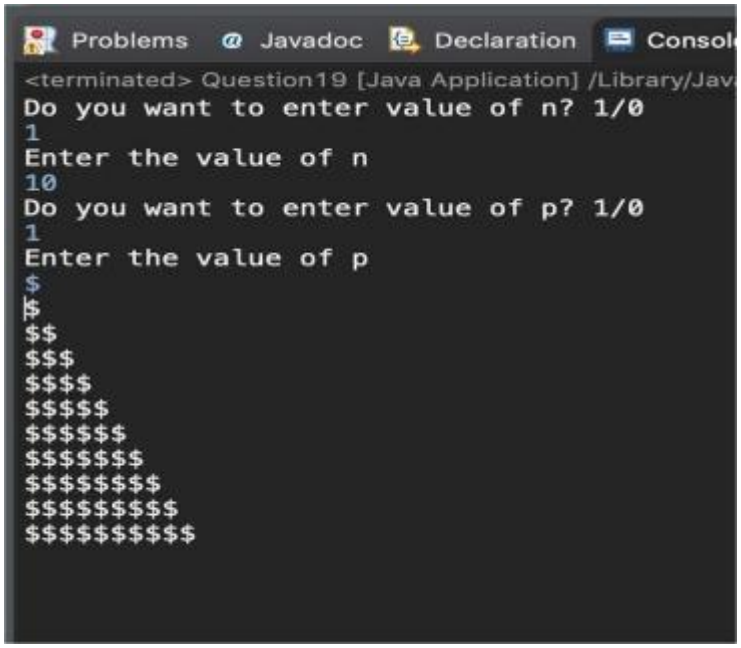
Code

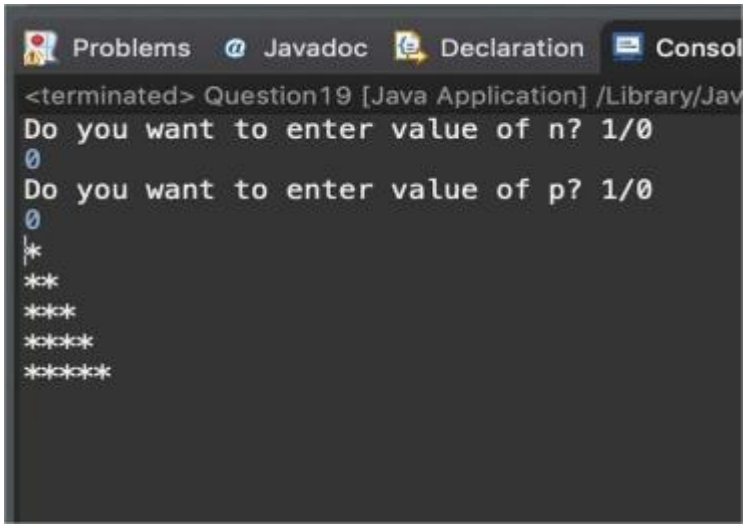
```
import java.util.*;
public class
Question19 {

        static void PRINT(char p, int n)
```

```
        {
                int i,j;
for(i=1;i<=n;i++)
                {
                        for(j=1;j<=i;j++)
         System.out.print(p);
                        System.out.println();
                }
        }
        static void PRINT(char p)
        {
        int i, j;

for(i=1;i<=5;i++)
                {
                        for(j=1;j<=i;j++)
         System.out.print(p);
                        System.out.println();
                }
        }
        static void PRINT(int n)
        {
        int i, j;

for(i=1;i<=n;i++)
                {
                        for(j=1;j<=i;j++)
         System.out.print('*');
                        System.out.println();
                }
        }
        static void PRINT()
        {
        int i, j;

for(i=1;i<=5;i++)
                {
                  for(j=1;j<=i;j++)
                System.out.print('*');
                        System.out.println();
                }
        }
        public static void main(String args[])
        {
  Scanner sc = new Scanner(System.in);   int n=0, ch;
                 boolean nc=false,np=false;
                char p='-';
```

```
                System.out.println("Do you want to enter value of n? 1/0");
                  ch=sc.nextInt();
if(ch==1)
                 {
                        System.out.println("Enter the value of
n");                    n=sc.nextInt();
        nc=true;
                 }
                System.out.println("Do you want to enter value of p? 1/0");
                  ch=sc.nextInt();
if(ch==1)
                 {
                        System.out.println("Enter the value of
p");                    p=sc.next().charAt(0);
                  np=true;
                 }
                  if(nc && np)
PRINT(p,n);
                 else if(nc && !np)
                        PRINT(n);
        else if(!nc && np)

        PRINT(p);       else
                  PRINT();



        }
}
```

Output





Question 20

Assume the following Doctor class definition is already available. Now, write a Java classes to allot patients for doctors in different departments such as Paediatric, ENT, and Dermatology depending on patient's choice. Test the functionalities of these classes using a Java program.

```
class Doctor  {
int doc_ID;    char
doc_name[30];
float yrs_exp;
```

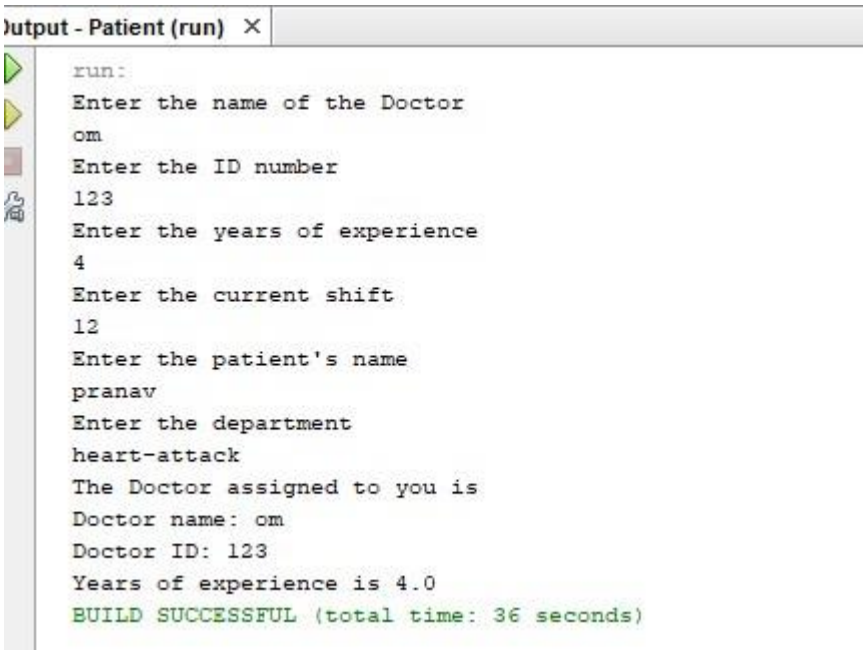
```
        char curr_shift[2];
    public:     void
    Read_data()
        {
        }           void
Write_data()
        {
        }
    }
```

The Code:

```
import java.util.*;
class Doctor
{
    int doc_ID;
    char
doc_name[];  float
yrs_exp;     char
curr_shift[];  String
dept;  void
Read_data()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the Doctor");

        doc_name = sc.next().toCharArray();
    System.out.println("Enter the ID number");
    doc_ID = sc.nextInt();
        System.out.println("Enter the years of experience");
    yrs_exp = sc.nextFloat();
        System.out.println("Enter the current shift");
        curr_shift = sc.next().toCharArray();

    }
    void Write_data()
    {
        System.out.println("Doctor name: "+new String(doc_name));
        System.out.println("Doctor ID: "+doc_ID);
        System.out.println("Years of experience is "+yrs_exp);

    }
}

public class Patient extends Doctor {
    String patient_name;
    String type_disease;
 //Paediatric, ENT, and Dermatology depending on patientâ€™s choice  void
getDetails()
    {
```

```
            Scanner sc = new Scanner(System.in);
        System.out.println("Enter the patient's name");
        patient_name = sc.nextLine();
        System.out.println("Enter the department");
        type_disease = sc.nextLine();
            Doctor ob = new Doctor();
            System.out.println("The Doctor assigned to you is ");
            Write_data();
        }
        public static void main(String args[])
        {
            Patient ob  = new Patient();
            ob.Read_data();
        ob.getDetails();
        }


}
```

Output:

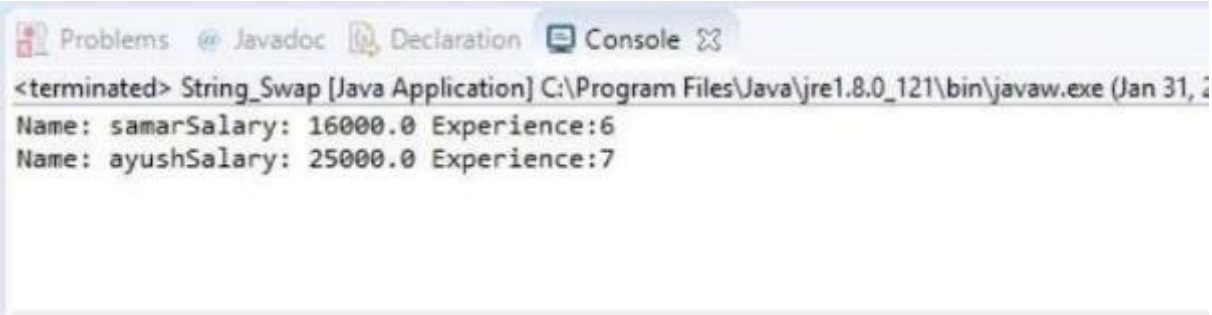


Question 21

With the existing abstract class ‘Employee’ derive another class called ‘Developer’ with required specifications. Create an array of ‘Java_Developer’ objects of size ‘n’. For the Java_Developers who have more than 5 years of experience give an increment of 10000INR. Write a Java program to test this and display the details of the employees who got the increment.

```
import java.util.*;
import java.lang.*;

abstract class Employee{
        String name;
float salary;     int
experience;     int
getExperience();
        float getsalary();
}

class Java_developer extends Employee{
Java_developer(String n, float s, int
exp){            name = n;      experience
= exp;
        if (exp>5) {
salary = s+ 5000;}
        else
 salary = s;
   }
        int getExperience() {
                return
experience;
        }
        float getsalary() {
                return salary;
        }
}

public class String_Swap {
        public static void main(String args[]) {
   Java_developer jv[] = new Java_developer[3]; // to test    jv[0]
=  new  Java_developer("Raj",5000,3);            jv[1]  =  new
Java_developer("samar",11000,6);
                jv[2] = new Java_developer("ayush",20000,7);

   for ( int i=0;i<3;i++) {     if
(jv[i].getExperience()>5) {
                                System.out.println("Name: "+ jv[i].name + "Salary: "+ jv[i].salary + "
Experience:" + jv[i].getExperience());

                        }
                }
        }
}
```

Output



Question 22

Assume you have a class ‘Vehicle’ with all basic information and a method to display its details. Using the class create new classes like Bike, Car, Bus, and Truck with their own specific information. A discount in the road tax is allowed for all the vehicle which are purchased in the year 2018 and later. Redefine the display method in the new classes to display updated details.

```
import java.io.*;
import java.util.*;
class Vehicle
{
   int maxSpeed = 120;
   String use="Petrol";

}


class Bike extends Vehicle
{
   String type="bike";
int wheels=2;
   String manufacturer="Honda";
   int year;
void
display()
   {
      System.out.println("Maximum Speed: " + super.maxSpeed+" use: "+super.use);
      System.out.println("type= "+type+" "+"wheels= "+wheels+" manufacturer=
"+manufacturer);       if(year>2018)
      {
            System.out.println("new car discount on road tax");
```

```
      }
else
{
            System.out.println("old car road tax");
      }
   }
}

class Car extends Vehicle
{
   String type="car";
int wheels=4;
   String manufacturer="Tesla";
int year;
   void display()
   {
      System.out.println("Maximum Speed: " + super.maxSpeed+" use: "+super.use);
      System.out.println("type= "+type+" "+"wheels= "+wheels+" manufacturer=
"+manufacturer);       if(year>2018)
      {
            System.out.println("new bike discount on road tax");
      }
else
      {
            System.out.println("old bike road tax");
      }
   }
}

class Truck extends Vehicle
{
   String type="truck";
int wheels=8;
   String manufacturer="TATA";
   int year;
void
display()
   {
      System.out.println("Maximum Speed: " + super.maxSpeed+" use: "+super.use);
      System.out.println("type= "+type+" "+"wheels= "+wheels+" manufacturer=
"+manufacturer);       if(year>2018)
      {
            System.out.println("new truck discount on road tax");
      }
else
      {
            System.out.println("old truck road tax");
```

```
    }
  }
}

class Bus extends Vehicle
{
  String type="bus";
int wheels=6;    String
manufacturer="Ashok
Layland";    int year;
  void display()
  {
    System.out.println("Maximum Speed: " + super.maxSpeed+" use: "+super.use);
    System.out.println("type= "+type+" "+"wheels= "+wheels+" manufacturer=
"+manufacturer);       if(year>2018)
    {
        System.out.println("new bus discount on road tax");
    }
else
    {
        System.out.println("old bus road tax");
    }
  }
}


class q21  {      public static void
main(String[] args)
  {
      Scanner in=new Scanner(System.in);
    Bus small = new Bus();
    System.out.println("enter the purchase
year");       small.year=in.nextInt();
small.display();

  }
}
```

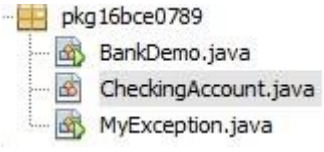
Output

```
enter the purchase year
2020
Maximum Speed: 120 use: Personal
type= bus wheels= 6 manufacturer= Ashok Layland
new bus discount on road tax
```

The Quesuions:

1. Implement a bank application where an alert message is issued when the minimum balance is going below 1000INR. Create an exception class and a Bank class for this application and test it with a Java program.

The Code:




```
package pkg16bce0789;

import java.util.Scanner;



public class BankDemo {  public

static void main(String [] args) {

    Scanner sc = new Scanner(System.in);
 CheckingAccount c = new CheckingAccount(101);
 int d = sc.nextInt();
 //System.out.println("Depositing $500...");
 c.deposit(d);


 try {
 System.out.println("Withdrawing $100...");
c.withdraw(100.00);
System.out.println("Withdrawing $600...");
c.withdraw(600.00);
 } catch (MyException e) {
 System.out.println("Sorry, but you are short $" + e.getAmount());
 e.printStackTrace();
 }
 }
 }
package pkg16bce0789;


class CheckingAccount {

private double balance;

private int number;
```

```
 public CheckingAccount(int number) {
this.number = number;
 }

 public void deposit(double amount) {
balance += amount;
 }

 public void withdraw(double amount) throws MyException {
if(amount >1000)
{
if(amount <= balance) {
balance -= amount;
 }else {
 double needs = amount - balance;
throw new MyException(needs);
 }
 }
 else
 {
   double needs = amount - balance;
throw new MyException(needs);
 }
 }

 public double getBalance() {
return balance;
 }

 public int getNumber() {
return number;
 }
 }
```

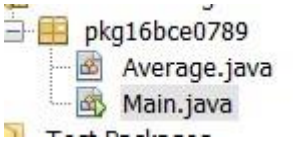
```
import java.io.*; class
MyException extends Exception {
private double amount; public
MyException(double amount) {
this.amount = amount;
 }
 public double getAmount() {
return amount;
 }
 }
```

The Output:

```
run:
600
Withdrawing $100...
Sorry, but you are short $-500.0
pkg16bce0789.MyException
        at pkg16bce0789.CheckingAccount.withdraw(CheckingAccount.java:33)
        at pkg16bce0789.BankDemo.main(BankDemo.java:19)
BUILD SUCCESSFUL (total time: 2 seconds)

run:
3600
Withdrawing $100...
Withdrawing $600...
BUILD SUCCESSFUL (total time: 6 seconds)
```

2. Develop a Java application for calculating the average mark of ‘n’ students. Read the number of courses they have appeared for the past semester and the marks in all the courses. The number of courses should not be zero and if it is zero handle that case with a standard exception.

The Code:




```
package pkg16bce0789;
/**
 *
```

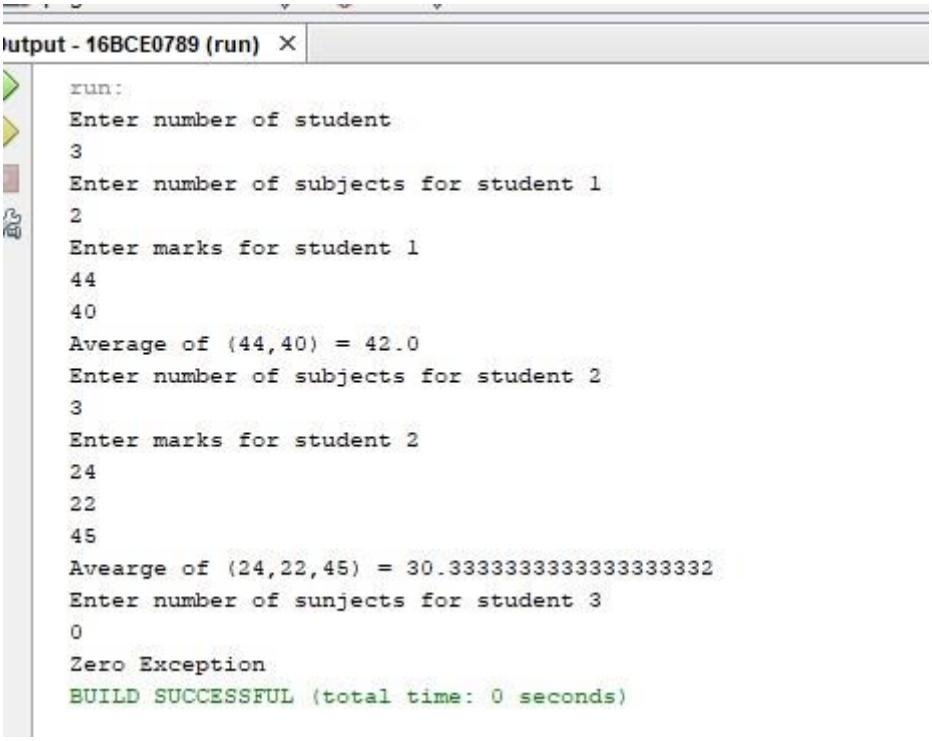
```
 * @author OM MISHRA
 */ class
Average

{
  double avg=0;
  void check (double amount) throws ZeroException
 {
   if (a==0)
       throw new ZeroException();
   else
         continue;
 }
  Average(int a[])
  {
     for(int  i=0;i<a.length;i++)
     {
       avg=avg+a[i];
     }
  }
```

```
}
class Main

{

  public static void main(String args[])

  {
int

i;

Sy

ste

m.

out

.pr

intl

n("

En

ter

nu

mb

er

of

su

bje

cts

");
```

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

int n=sc.nextInt();

int[] a=new int[n];

System.out.println("Enter marks");

for( i=0;i<n;i++)

{
  a[i]=sc.nextInt();

}

Average c = new Average(a);
```

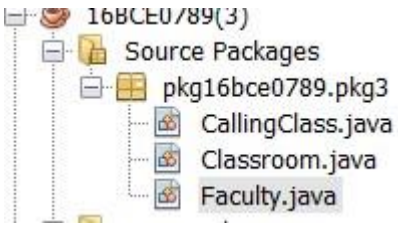
```
    c.check();

    System.out.print("Average of (");




    for(i=0;i<n-1;i++)

    {

     System.out.print(a[i]+",");

    }
    System.out.println(a[i]+") ="+c.avg/n);




  }

}
```

The Output:



3. Write a Java program to develop an application where you have a class ‘ClassRoom’ which have to be used by the (thread) objects of class ‘Faculty’ to deliver their course contents. Since the ‘Faculty’ class objects are active simultaneously, synchronize the usage of the object of ‘Classroom’.

The Code:

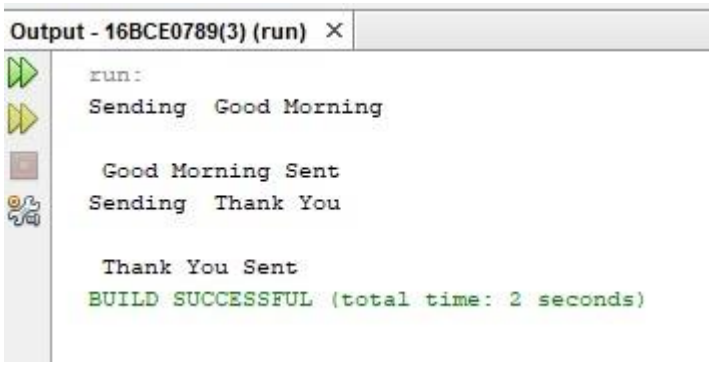



```
// A Class used to send a message

class Faculty

{
   public void send(String msg)
   {
      System.out.println("Sending\t"  + msg );
      try
      {
         Thread.sleep(1000);
      }
      catch (Exception e)
```

```
        {
            System.out.println("Thread  interrupted.");
        }
        System.out.println("\n" + msg + "Sent");
    }
}
package pkg16bce0789.pkg3;

/**
 *
 * @author OM MISHRA
 */ class Classroom extends

Thread

{
    private String msg;

private Thread t;

    Faculty  faculty;


    // Recieves a message object and a string
    // message to be sent
    Classroom(String m,  Faculty obj)
    {
        msg = m;

faculty = obj;

    }


    public void run()
    {
        // Only one thread can send a message
        // at a time.
        synchronized(faculty)
        {
            // synchronizing the snd object             faculty.send(msg);
        }
    }
}
package pkg16bce0789.pkg3;


/**
 *
 * @author OM MISHRA
```

```
 */ class

CallingClass

{
   public static void main(String args[])
   {
      Faculty f1 = new Faculty();

Classroom C1 =             new Classroom(

" Good Morning " , f1 );         Classroom

C2 =             new Classroom( " Thank

You " , f1 );



      // Start two threads of ThreadedSend type
      C1.start();
      C2.start();


      // wait for threads to end
      try
      {
         C1.join();

C2.join();

      }
      catch(Exception e)
      {
         System.out.println("Interrupted");
      }
   }
}
```

The Output:



4. We need to train the patient data to predict whether a new patient may or may not get the disease. Implement a data training application in Java which prepares the data set used in a disease prediction algorithm. The ‘TraingData’ class has an array of records (patient-id, patientage, diseaseid, and disease-seriousness-index) and two methods ‘writeData’ and ‘readData’. In a multi-threaded environment, if one thread is writing the data then other threads have to wait and if one thread is reading the data then other threads have to wait.

The Code:

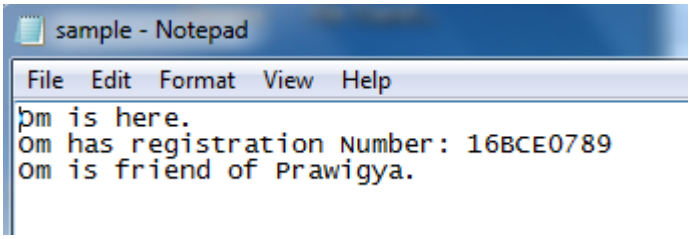

```
package pkg16bce0789.pkg4;


/**
 *
 * @author OM MISHRA
 */ import

java.io.*;

import

java.util.*;

import

java.lang.*;


public class Multithread
{
   public static void main(String[] args)
   {


      Scanner sc = new

Scanner(System.in);        //int n = 8;

// Number of threads         int n =

sc.nextInt();        for (int i=0; i<n;

i++)

      {
```

```
        TrainingData object = new TrainingData();
object.writeData();
        object.readData();
object.start();
      }
   }
}
package pkg16bce0789.pkg4;

/**
 *
 * @author OM MISHRA
 */ import
java.io.*;
import
java.util.*;
import
java.lang.*;

class TrainingData extends Thread
{
   public int
Patientid[];
public int age[];
public int disid[];
public int dsi[];
static int i = 0;

   public void writeData()
   {
      Patientid = new int[]
{1,2,3,4,5,6,7,8,9};        age = new int[]
```

```
{12,56,45,89,26,78,24,67,74};         disid =

new int[] {1,2,1,1,2,2,1,1,1};         dsi = new

int[] {1,2,3,1,1,3,2,1,2};

   }


   public void readData()
   {
       while(i!=9)
          {
          System.out.println("Pateint Id :"+Patientid[i]);
          System.out.println("Pateint age :"+age[i]);
          System.out.println("Disease Id :"+disid[i]);
          System.out.println("disease-seriousness-index :"+dsi[i]);
          i++;
          }
   }




   public void run()
   {

try

{



          // Displaying the thread that is running
          int n =(int) Thread.currentThread().getId();
          System.out.println ("Thread " + n + " is running");
          System.out.println("Pateint id "+Patientid[(n-11)]);
          System.out.println("Patient age "+age[(n-11)]);
          System.out.println("Disease id "+disid[(n-11)]);
          System.out.println("disease-seriousness-index "+dsi[(n-11)]);


       }
       catch (Exception e)
       {
          // Throwing an exception
          System.out.println ("Exception is caught");
       }
   }
}
```

The Output:

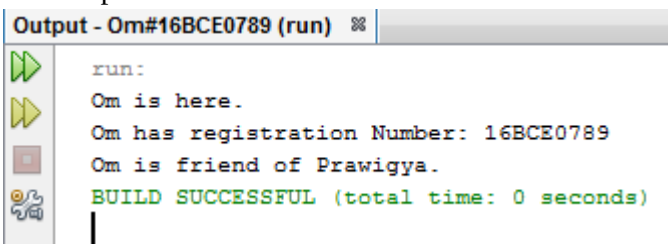
5. Write a Java program to read the contents in a text file and display in the console.
Answer:
The Sample Text:



The Code:

```
package om.pkg16bce0789;

/**
 *
 * @author 16BCE0789
 */
import java.io.File;
import java.util.Scanner;
import java.lang.*;
public class Om16BCE0789 {


    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws Exception{
        File file = new
File("C:\\Users\\16BCE0789\\Documents\\NetBeansProjects\\Om#16BCE0789\\sample.txt");
        Scanner sc = new Scanner(file);

        //BufferedReader br = new BufferedReader(new FileReader(file));

        String st;
        while(sc.hasNextLine())
            System.out.println(sc.nextLine());
    }

}
```

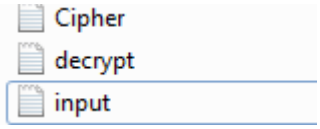
The Output:



6. Write a Java program to implement Caesar's cipher to encrypt the data stored in a text file named, 'input.txt'. The encrypted data should be stored in 'cipher.txt'. Also decrypt the contents in 'cipher.txt' and store the plain text in 'plain.txt'.

Answer:

The Sample:

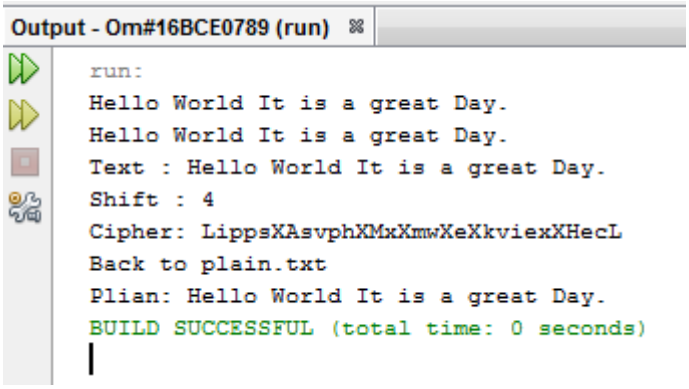


The Code:

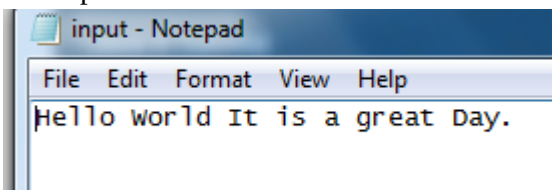
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package om.pkg16bce0789;

/**
 *
 * @author 16BCE0789
 */
import java.io.File;
import java.util.Scanner;
import java.lang.*;
public class Om16BCE0789 {


    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws Exception{
        File file = new
File("C:\\Users\\16BCE0789\\Documents\\NetBeansProjects\\Om#16BCE0789\\input.txt");
        Scanner sc = new Scanner(file);


        String st="Hello World It is a great Day.";
        while(sc.hasNextLine())
        {

            System.out.println(sc.nextLine());
            st = st+sc.nextLine();
        }
        System.out.println(st);
        int s = 4;
        System.out.println("Text : "+st);
        System.out.println("Shift : "+s);
        System.out.println("Cipher: "+ encrypt(st,s));
        System.out.println("Back to plain.txt");
        System.out.println("Plian: "+ st);
    }

        public static StringBuffer encrypt(String st, int s)
```

```
    {
        StringBuffer result = new StringBuffer();

        for(int i=0;i<st.length();i++)
        {
            if(Character.isUpperCase(st.charAt(i)))
            {
            char ch = (char) (((int)st.charAt(i) + s - 65) % 26 + 65);
            result.append(ch);
            }
            else
            {
                char ch = (char) (((int)st.charAt(i) + s - 97) % 26 + 97);
                result.append(ch);
            }
        }
        return result;
    }
}
```

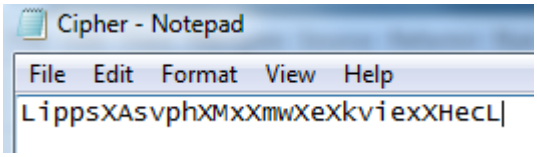
The Output:



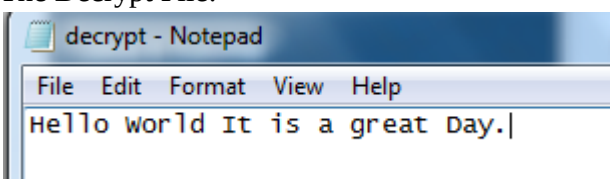
The Input File:



The Cipher File:



The Decrypt File:



7. Write a Java program to serialize the objects of the class ‘Player’ which has the following details: player_id, player_name, age, team_name, run_rate, wickets. Use appropriate classes for writing and reading the data into and from files.

Answer:

The Sample:

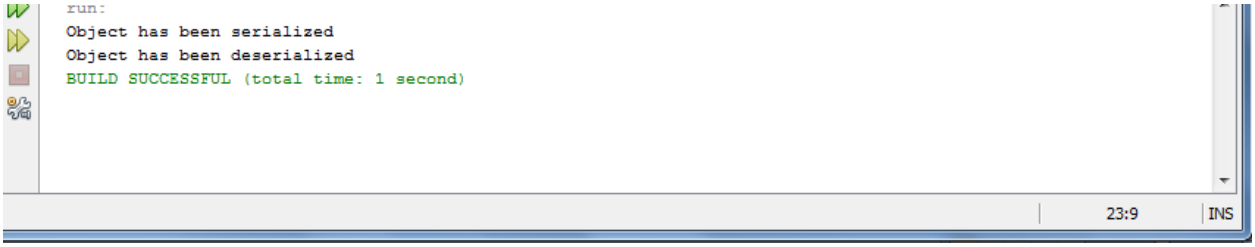
```
package pkg16bce0892;
import java.io.*;

public class Player implements java.io.Serializable
{
	int player_id;
	String name;
	int age;
	String team_name;
	double run_rate;
	int wickets;

	//Constructor
	public Player(int player_id, String name, int age, String team_name, double run_rate, int wickets)
	{
	   this.player_id = player_id;
	   this.name = name;
	   this.age = age;
	   this.team_name = team_name;
	   this.run_rate = run_rate;
	   this.wickets = wickets;

	}
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pkg16bce0789;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

/**
 *
 * @author 16BCE0789
 */
public class Read
{
   public static void main(String args[])
   {
```

```
        String filename = "abc.txt";
        //Serialization
        try
        {
            FileOutputStream file = new FileOutputStream(filename);
            ObjectOutputStream out = new ObjectOutputStream(file);

            out.writeObject(object);

            out.close();
            file.close();

            System.out.println("File has been serialized");
        }
        catch(IOException ex)
        {
            System.out.println("IOException is caught");
        }
        Player object1 = null;

        //Deserialization
        try
        {
            FileInputStream file = new FileInputStream(filename);
            ObjectInputStream in = new ObjectInputStream(file);

            object1 = (Player)in.readObject();
            in.close();
            file.close();

            System.out.println("Object has been deserialized");
        }
        catch(IOException ex)
        {
            System.out.println("IOExceotion caught");
        }
        catch(ClassNotFoundException ex)
        {
            System.out.println("ClassNotFound");

        }

    }

}
```

The Code:

The Output:



8. A double-ended queue is a data structure in which the enqueue and dequeue operations can be done at both the ends of the queue. Implement a doubleended queue using a suitable Java Collection object. Develop a Java application to test the same.
The Answer:
The Code:

```
package DoublyLinkedList;

/**
 *
 * @author OM MISHRA
 */
public class DoublyLinkedList<T> {

private Node<T> head;
private Node<T> tail;

/**
 * Inserts the value at the first position (head) of LinkedList.
 *
 * @param value
 *          to be inserted
 */
public void insertFirst(final T value) {
        final Node<T> node = new Node<>(value);
        node.next = head;
        if (head != null) {
                head.previous = node;
        }

        head = node;

        if (tail == null) {
                tail = node;
        }
}

/**
 * Inserts the value at the last position (tail) of LinkedList.
 *
 * @param value
```

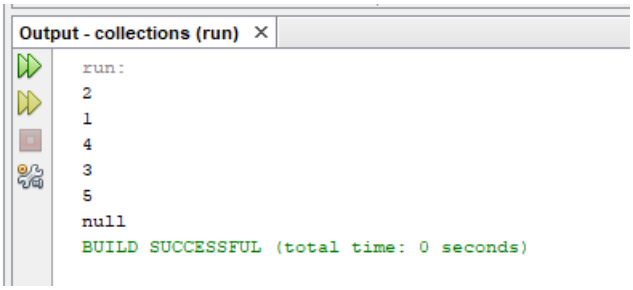
```
 */
public void insertLast(final T value) {
	final Node<T> node = new Node<>(value);
	if (tail != null) {
		tail.next = node;
		node.previous = tail;
	}
	tail = node;

	if (head == null) {
		head = node;
	}
}

/**
 * Removes the node from first position (head) of LinkedList.
 *
 * @return the value of node deleted. Null if no nodes are present
 */
public T removeFirst() {
	T value = null;
	if (head != null) {
		value = head.value;
		if (head == tail) {
			tail = null;
		}

		head = head.next;
		head.previous = null;
	}

	return value;
}

/**
 * Removes the node from last position (tail) of LinkedList.
 *
 * @return the value of node deleted. Null if no nodes are present
 */
public T removeLast() {
	T value = null;
	if (tail != null) {
		value = tail.value;

		if (tail == head) {
			head = tail = null;
		} else {
```

```
                        tail = tail.previous;
                        tail.next = null;
                }
        }
        return value;
}

/**
 * Removes the first occurance of node having the value same as input value.
 *
 * @param value
 *          to be removed
 * @return deleted node's value if node is found else null;
 */
public T remove(final T value) {
        T deletedObj = null;
        if (head != null) {
                if (head == tail) {
                        if (head.value.equals(value)) {
                                deletedObj = head.value;
                                head = tail = null;
                        }
                } else {
                        Node<T> node = head;
                        do {
                                if (node.value.equals(value)) {
                                        deletedObj = node.value;

                                        if (node.previous != null) {
                                                node.previous.next = node.next;

                                        }
                                        node.next.previous = node.previous;
                                        break;
                                }
                                node = node.next;
                        } while (node != null);
                }
        }

        return deletedObj;
}

/**
 * Implementation of a Node of a Doubly Linked List.
 *
 * @author Sain Technology Solutions
```

```
 *
 * @param <T>
 */
private static class Node<T> {
	T value;
	Node<T> next;
	Node<T> previous;

	private Node(T value) {
		this.value = value;
	}

	@Override
	public String toString() {
		return "Node [value=" + value + "]";
	}

}

/**
 * Entry point for testing LinkedList.
 */
public static void main(String[] args) {
	final DoublyLinkedList<Integer> doublyLinkedList = new DoublyLinkedList<>();

	// Inserts the node with value 5 at the head position
	doublyLinkedList.insertFirst(5);

	// Inserts the node with value 1 at the head position, pushing the
	// previously inserted node to second position
	doublyLinkedList.insertFirst(1);

	// Inserts the node with value 2 at the head position, pushing the
	// previously inserted node to second position
	doublyLinkedList.insertFirst(2);

	// Inserts the node with value 3 at the tail position
	doublyLinkedList.insertLast(3);

	// Inserts the node with value 4 at the tail position, pushing the
	// previously inserted node to second position from last
	doublyLinkedList.insertLast(4);

	// At this point, LinkedList will look like: 2 <=> 1 <=> 5 <=> 3 <=> 4

	// Removes the node with value 2 since it is head node. This operation
	// will also make node with value 1 as head node
```

```
        System.out.println(doublyLinkedList.removeFirst());

        // Removes the node with value 1 since it is head node. This operation
        // will also make node with value 5 as head node
        System.out.println(doublyLinkedList.removeFirst());

        // Removes the node with value 4 since it is tail node. This operation
        // will also make node with value 3 as tail node
        System.out.println(doublyLinkedList.removeLast());

        // Removes the node with value 3 since it is tail node. This operation
        // will also make node with value 5 as tail node
        System.out.println(doublyLinkedList.removeLast());

        // Removes the node with value 5
        System.out.println(doublyLinkedList.remove(5));

        // Returns null since there is no node with value 2 as it was removed
        // due to earlier removeXXX method calls
        System.out.println(doublyLinkedList.remove(2));

}

}
```

The Output:



9. Write a Java application to store the different comments given by the users on the surgical attack by our Indian army on terrorist camps in Pakistan held on 26-Feb-2019, 3.30am. The comment should be of ONE word and the duplicates need not to be stored. Use suitable Java Collection object to implement this program and test it.

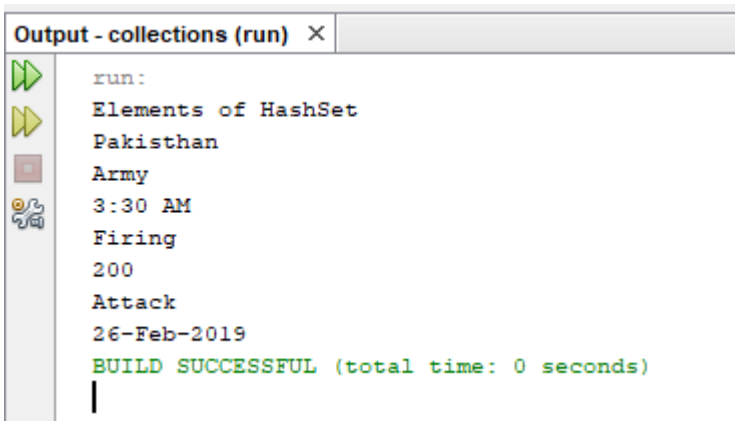
The Answer:

The Code:

```
package Collections;
import java.util.*;
/**
 *
 * @author OM MISHRA
 */
public class hashset {
   public static void main(String arg[]){
      HashSet hs=new HashSet();
```

```
        hs.add("Pakisthan");
        hs.add("3:30 AM");
        hs.add("26-Feb-2019");
        hs.add("Army");
        hs.add("Attack");
        hs.add("Army");
        hs.add("Army");
        hs.add("Firing");
        hs.add(200);
        Iterator it1=hs.iterator();
        System.out.println("Elements of HashSet");
        while(it1.hasNext())
            System.out.println(it1.next());
    }
}
```

The Output:



10. Implement a Lucky Draw game in which the user has to input an integer ‘input’ and multiply it with a random integer to get the product ‘p’. Depending upon the value of ‘p’ display the prize amount. Use a suitable Java Collection object to store the prize amount for each value of ‘p’. [Hint: Use modulo operation to get final value of ‘p’]

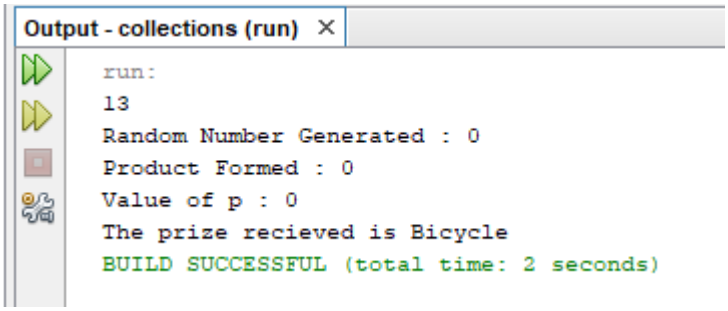
The Answer:

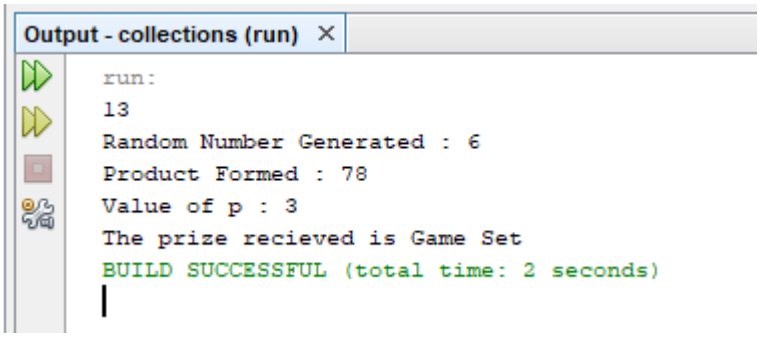
The Code:

```
package DoublyLinkedList;
import java.util.*;
/**
 *
 * @author OM MISHRA
 */
public class Linkedhashmap {
    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Random rn = new Random();
        int n1 = rn.nextInt(50);
```

```
        System.out.println("Random Number Generated : "+n1);
        int val = n*n1;
        System.out.println("Product Formed : "+val);
        int p = val%5;
        System.out.println("Value of p : "+p);
        LinkedHashMap<Integer,String> hm=new LinkedHashMap();
        hm.put(0,"Bicycle");
        hm.put(1,"Car");
        hm.put(2,"Bike");
        hm.put(3,"Game Set");
        hm.put(4, "Dog");
        Set set=hm.entrySet();
        Iterator itr=set.iterator();
        while(itr.hasNext()){
         //Converting to Map.Entry so that we can get key and value separately
         Map.Entry entry=(Map.Entry)itr.next();
         if(p == (int)(entry.getKey()))
         {
            System.out.println("The prize recieved is "+entry.getValue());
         }
      }
    }
  }
```

The Output:





1. Write a generic Java method to print the user input of any type (userdefined object also…) for ‘n’ number of times. Write a Java program to test it.

The answer:

The

Code:

```
/*

* To change this license header, choose License
  Headers in Project Properties.

* To change this template file, choose Tools |
  Templates

* and open the template in the editor.

 */ package

genericprinting;



/**
 *
* @author 16BCE0789

 */ import java.util.*;

import java.io.*; import

java.lang.*; public class

GenericPrinting <T>

{     int

objvar;

public

static

<E>

void

printAr

ray(E[]

inputA

rray)

    {
```

```
        //Display Array
Elements        for(E
element: inputArray)
        {
            System.out.printf("%s ", element);
        }
        System.out.println();
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int
n,i;
int ch=1;
do{
        System.out.println("Enter 1 for Integer
Array");
        System.out.println("Enter 2 for Character
Array");
        System.out.println("Enter 3 for Double
Array");
        System.out.println("Enter 4 for String Array");
        System.out.println("Enter 5 for Object
Array");         System.out.println("Enter 6 to
exit");         ch = sc.nextInt();

        switch(ch)
        {
case 1:
                System.out.println("Enter the size of
array");

                n = sc.nextInt();
                System.out.println("Enter elements");
                Integer[] intArray = new Integer[n];
```

```
            for(i=0;i<n;i++)
            {
               intArray[i] = sc.nextInt();
            }
         System.out.println("Displaying Integer
Array");          printArray(intArray);
break;          case 2:
            System.out.println("Enter the size of
array");

            n = sc.nextInt();
            System.out.println("Enter elements");
            Character[] charArray = new
Character[n];


            for(i=0;i<n;i++)
           {
               charArray[i] = sc.next().charAt(0);
            }
         System.out.println("Displaying Character
Array");          printArray(charArray);
break;          case 3:
            System.out.println("Enter the size of
array");             n = sc.nextInt();
            System.out.println("Enter elements");
            Double[] doubleArray = new Double[n];

            for(i=0;i<n;i++)
            {
               doubleArray[i] = sc.nextDouble();
            }
         System.out.println("Displaying Double
Array");          printArray(doubleArray);
         break;
```

```
        case 4:
           System.out.println("Enter the size of
array");              n = sc.nextInt();
           System.out.println("Enter elements");
           String[] stringArray = new String[n];
          sc.nextLine();
           for(i=0;i<n;i++)
           {
              stringArray[i] = sc.nextLine();
           }
        System.out.println("Displaying String
Array");           printArray(stringArray);
break;          case 5:
           System.out.println("Enter the size of
array");              n = sc.nextInt();
           System.out.println("Enter elements");
           GenericPrinting[] obj = new
GenericPrinting[n];

           for(i=0;i<n;i++)
           {
              obj[i] = new GenericPrinting();
obj[i].objvar = sc.nextInt();
           }
           System.out.println("Displaying Object
Array");              printArray(obj);
break;             case 6:              break;
default:
          System.out.println("Invalid Input");
           break;
      }

      }while(ch!=6);
```

```
    }
}
```

The Output:

```
run:
Enter 1 for Integer Array
Enter 2 for Character Array
Enter 3 for Double Array
Enter 4 for String Array
Enter 5 for Object Array
Enter 6 to exit
1
Enter the size of array
4
Enter elements
12
35
67
78
Displaying Integer Array
12 35 67 78

Enter 1 for Integer Array
Enter 2 for Character Array
Enter 3 for Double Array
Enter 4 for String Array
Enter 5 for Object Array
Enter 6 to exit
2
Enter the size of array
4
Enter elements
o
l
m
k
Displaying Character Array
o l m k
```

```
Enter 1 for Integer Array
Enter 2 for Character Array
Enter 3 for Double Array
Enter 4 for String Array
Enter 5 for Object Array
Enter 6 to exit
3
Enter the size of array
4
Enter elements
6.7
6.34
7.0
3.8
Displaying Double Array
6.7 6.34 7.0 3.8
```

```
Enter 1 for Integer Array
Enter 2 for Character Array
Enter 3 for Double Array
Enter 4 for String Array
Enter 5 for Object Array
Enter 6 to exit
4
Enter the size of array
4
Enter elements
om
tom
gom
kom
Displaying String Array
om tom gom kom
```

```
Enter 1 for Integer Array
Enter 2 for Character Array
Enter 3 for Double Array
Enter 4 for String Array
Enter 5 for Object Array
Enter 6 to exit
5
Enter the size of array
4
Enter elements
1
34
6
90
Displaying Object Array
genericprinting.GenericPrinting@1f96302 genericprinting.GenericPrinting@14eac69 genericprinting.GenericPrinting
```

```
Enter 1 for Integer Array
Enter 2 for Character Array
Enter 3 for Double Array
Enter 4 for String Array
Enter 5 for Object Array
Enter 6 to exit
6
BUILD SUCCESSFUL (total time: 53 seconds)
```

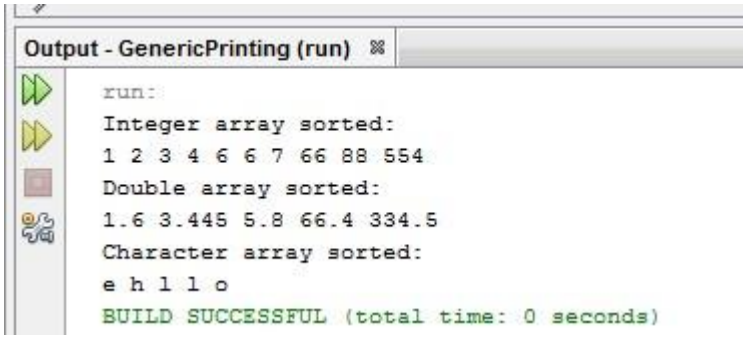
2. Write a Java program to sort an integer array, a double array and a character array using a generic method.

The answer:

The Code:

```
package genericprinting;


/**
 *
 * @author 16BCE0789
 */ import java.util.*;

import java.io.*; import

java.lang.*; public class

GenericPrinting <T>

{
  public static <E> void

sortArray(E[] ipAr ){

Arrays.sort(ipAr);     for (E element:

ipAr){

   System.out.print(element + " ");
   }
   System.out.println();
   }


   public static void main(String args[]){
   Integer[] inArr = {1,2,3,7,6,4,88,6,554,66};
   Double[] dbArr = {1.6,5.8,3.445,334.5,66.4};
   Character[] charArr = {'h','e','l','l','o'};
   System.out.println("Integer array sorted:");

sortArray(inArr);

   System.out.println("Double array sorted:");

sortArray(dbArr);

   System.out.println("Character array sorted:");

sortArray(charArr);

   }
}
```

The Output:



3. Design a generic stack data structure which can handle integer, double, character and any userdefined objects. Write a menu-driven Java program to test it.
The answer:
The Code:

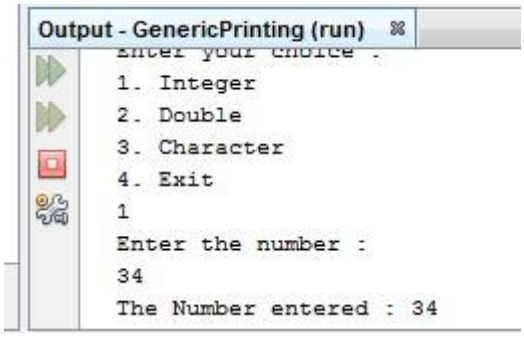
```
package genericprinting;


/**
 *
 * @author 16BCE0789
 */ import java.util.*;

import java.io.*; import

java.lang.*; public class

GenericPrinting <T>

{
   public static <E> void printArray(E[]
inputArray)
   {
      //Display Array

Elements        for(E

element: inputArray)

      {
         System.out.printf("%s ", element);
      }
      System.out.println();
   }
   private ArrayList<T> stack = new ArrayList<T>

();     private int top = 0;
```

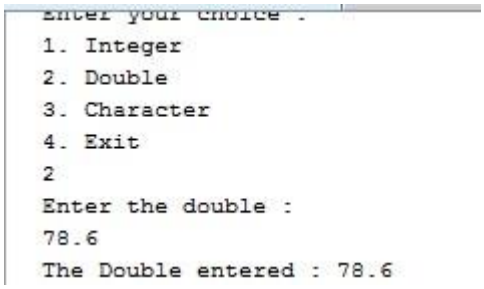
```
   public int size (){return top;}

   public void push(T item){

stack.add(top++, item);

   }


   public T pop(){

return stack.remove(--

top);

   }


    public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);


      Integer[] intArray = {1,2,3,4,5};
      Double[] doubleArray = {1.1,2.2,3.3,4.4,5.5};
      Character[] charArray = {'s','t','a','r','t'};
      System.out.println("Displaying Integer

Array");        printArray(intArray);

      System.out.println("Displaying Double

Array");        printArray(doubleArray);

      System.out.println("Displaying Character

Array");        printArray(charArray);

      int

ch;

do

      {
      System.out.println("Enter your choice : ");
System.out.println("1. Integer");
      System.out.println("2. Double");
      System.out.println("3. Character");
      System.out.println("4. Exit");
```

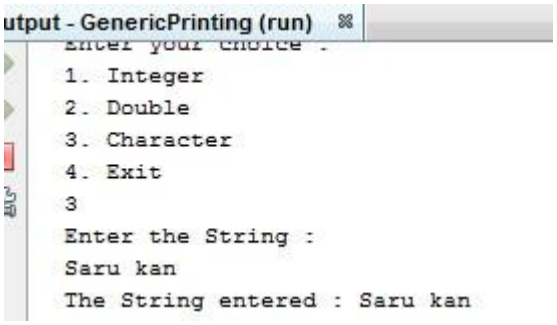
```
        ch = sc.nextInt();


        switch(ch)
        {
case 1:
                GenericPrinting<Integer> s = new
GenericPrinting<Integer> ();
System.out.println("Enter the number : ");                int f =
sc.nextInt();


s.push(f);
int i = s.pop();
                System.out.println("The Number entered
: "+i);
                break;


            case 2:
                GenericPrinting<Double> s1 = new
GenericPrinting<Double> ();
System.out.println("Enter the double : ");                double f1 =
sc.nextDouble();                s1.push(f1);                double i1 =
s1.pop();
                System.out.println("The Double entered :
"+i1);
                break;


            case 3:
                GenericPrinting<String> s2 = new GenericPrinting<String> ();
                sc.nextLine();
                System.out.println("Enter the
String : ");                String f2 =
sc.nextLine();                s2.push(f2);
                String i2 = s2.pop();
```

```
            System.out.println("The String entered :
"+i2);
            break;


        case 4:

break;



        default:
            System.out.println("Wrong Choice");
            break;
      }
      }while(ch!=4);


   }
}
```

The Output:

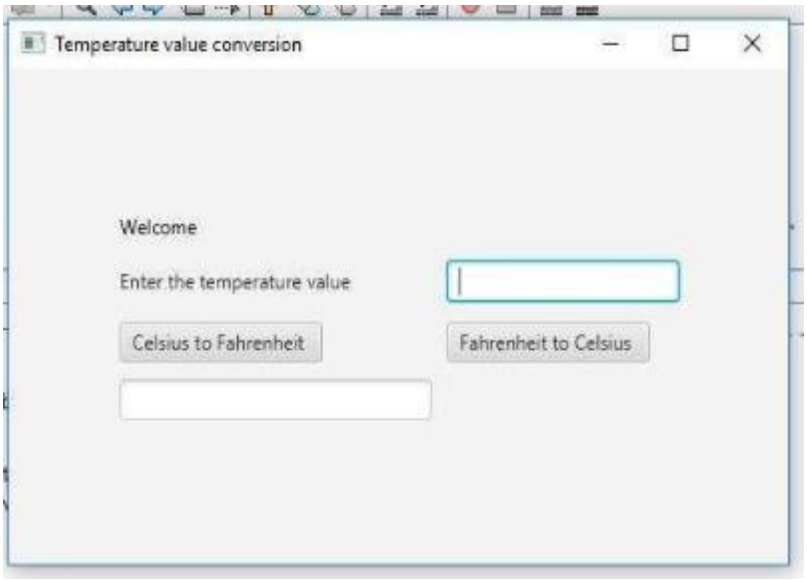






```
Enter your choice :
1. Integer
2. Double
3. Character
4. Exit
5
Wrong Choice
Enter your choice :
1. Integer
2. Double
3. Character
4. Exit
4
```

1. Design a GUI application using JavaFX classes to convert the temperature in Celsius to Fahrenheit and vice versa.
   [Hint: Use the following UI design as a reference.]



Answer:

The Code:

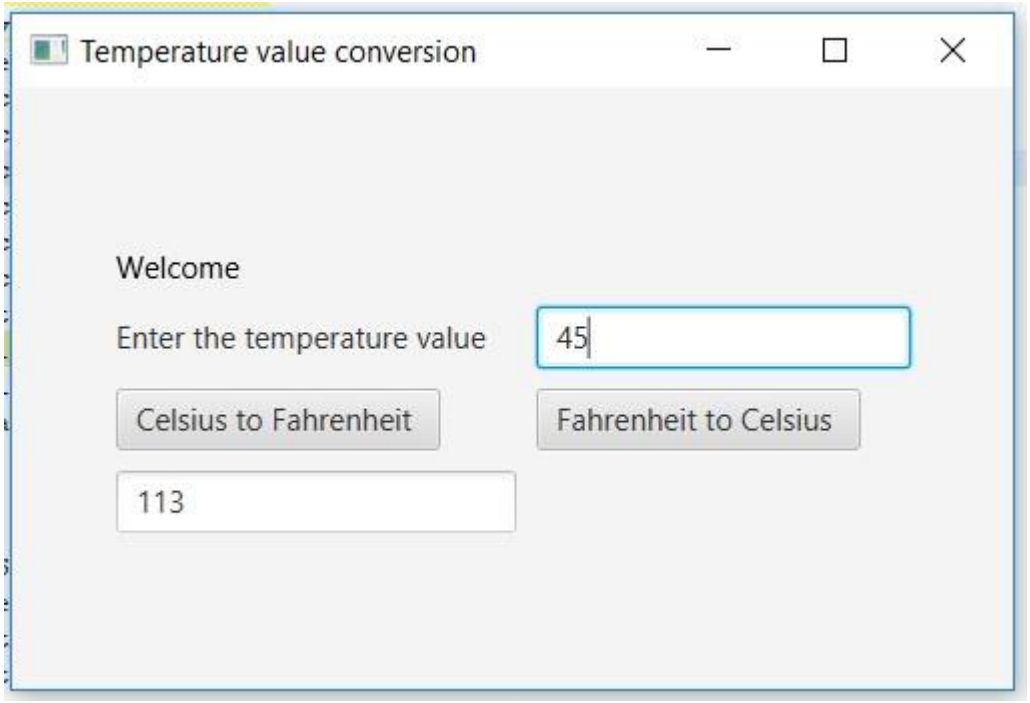
```
/*
* To change this license header, choose License Headers in Project
  Properties.  * To change this template file, choose Tools | Templates
  * and open the template in the editor.
 */
package pkg16bce0789_3;

/**
 *
* @author OM MISHRA
```

```
 */

//package frames;
import javafx.geometry.Insets; import
javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos; import
javafx.scene.Scene; import
javafx.scene.control.Button; import
javafx.scene.control.Label; import
javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text; import
javafx.stage.Stage; public class Main
extends Application{     public static
void main(String[] args) {
        launch(args);
    }
@Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Temperature value conversion");
GridPane grid = new GridPane();
grid.setAlignment(Pos.CENTER);
        grid.setHgap(10);
grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));
        Text scenetitle = new Text("Welcome");
        Label promt = new Label("Enter the temperature value");
        promt.setPrefSize(200, 30);
TextField tempVal = new TextField();
        String temp = String.ValueOf(tempVal);
        Button btn1 = new Button("Celsius to Fahrenheit ");
        Button btn2 = new Button("Fahrenheit to Celsius
");         TextField res = new TextField();
grid.add(scenetitle, 0, 0);          grid.add(promt, 0, 1);
grid.add(tempVal,1, 1);         grid.add(btn1, 0, 2);
grid.add(btn2, 1, 2);
        grid.add(res, 0, 3);

        String s1 = btn1.temp;
double c = Double.parseDouble(s1);
double f = c*(9/5)+32;          String
result = String.valueOf(f);
        res = new TextField(result);
```

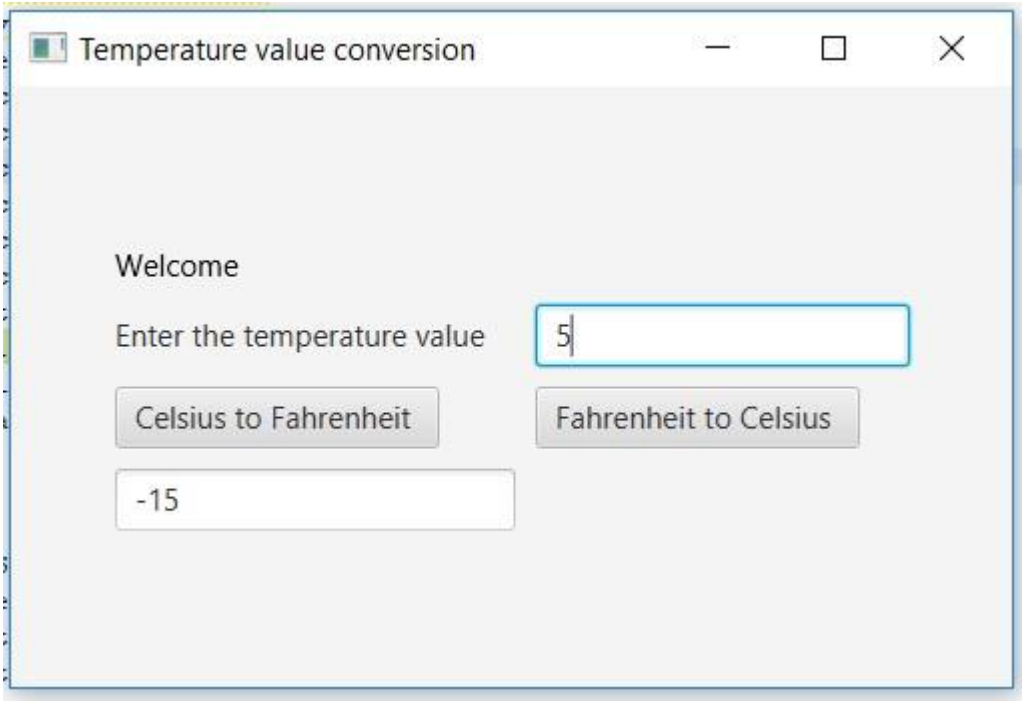
```
        String s2 = btn2.temp;
double f = Double.parseDouble(s2);
double c = (f - 32)*(5/9);
        String result = String.valueOf(f);
        res = new TextField(result);

        Scene scene = new Scene(grid, 500, 300);
primaryStage.setScene(scene);
primaryStage.show();
    }
```

The Output:

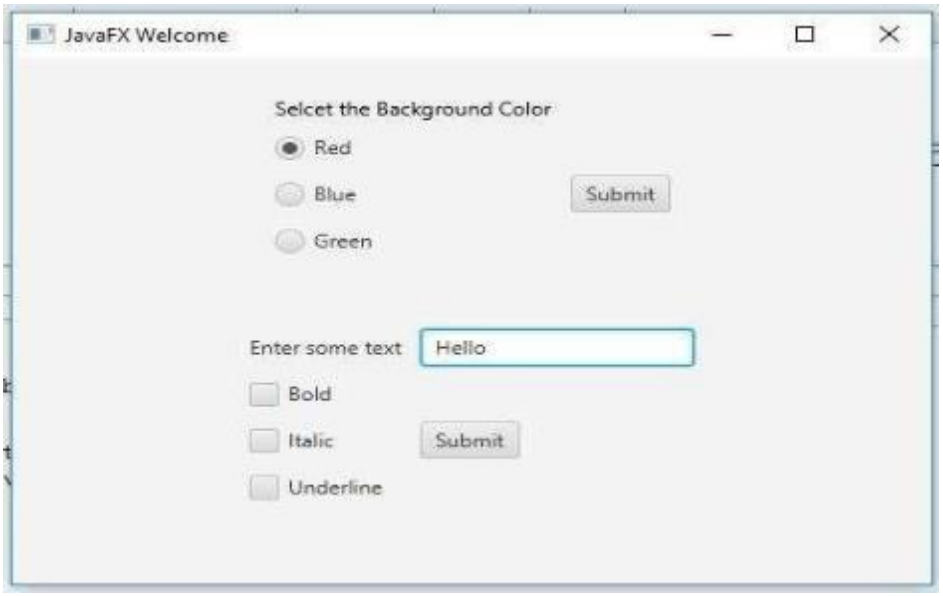
On clicking Celsius to Fahrenheit:-



On clicking Fahrenheit to Celsius



2. Design a GUI application using JavaFX classes to change the background color of the application using RadioButton class objects and to change the text appearance by using CheckBox class objects. Use two pane layout for each operation.
[Hint: Use the following UI design as a reference.]



The Code:

```
package pkg16bce0789_3;

/**
```

```
 *
 * @author OM MISHRA
 */

import javafx.geometry.Insets; import
javafx.application.Application; import
javafx.event.ActionEvent; import
javafx.event.EventHandler; import
javafx.geometry.Pos; import
javafx.scene.Scene; import
javafx.scene.control.Button; import
javafx.scene.control.Label; import
javafx.scene.control.TextField; import
javafx.scene.layout.GridPane; import
javafx.scene.text.Text; import
javafx.stage.Stage; import
javafx.application.Application; import
javafx.scene.Scene; import
javafx.scene.control.RadioButton;
import javafx.scene.layout.HBox;
import javafx.stage.Stage; import
javafx.application.Application; import
javafx.scene.Scene; import
javafx.scene.control.CheckBox; import
javafx.scene.layout.HBox;
import javafx.stage.Stage;



public class Main extends Application  {


   @Override
   public void start(Stage primaryStage) throws Exception {
primaryStage.setTitle("Welcome JavaFX");
       Label promt = new Label("Select the Backgroud Colour");

       RadioButton radioButton1 = new RadioButton("Red");

       RadioButton radioButton2 = new RadioButton("Blue");
       RadioButton radioButton3 = new RadioButton("Green");
       RadioButton radioButton4 = new RadioButton("Yellow");

       Button button = new Button("Submit");
```

```
        Label promt1 = new Label("Enter the some text");
        promt.setPrefSize(200, 30);
        TextField tex = new TextField();

        CheckBox ch1 = new CheckBox("Bold");
        CheckBox ch2 = new CheckBox("Italics");
        CheckBox ch3 = new CheckBox("Underline");
        Button button1 = new Button("Submit");

        HBox hbox = new HBox(promt,radioButton1, radioButton2, radioButton3,
radioButton4,button,promt1,tex,ch1,ch2,ch3,button1);
Scene scene = new Scene(hbox, 1200, 500);
primaryStage.setScene(scene);
primaryStage.show();

    }

    public static void main(String[] args) {
        Application.launch(args);
    }

}
```

The Output:

hello

3. Design a menu driven GUI application using JavaFX classes to perform arithmetic and string operations such as,

    a. Numerical data

        i. Addition ii. Subtraction iii. Multiplication iv. Division

    b. String

        i. Finding length ii. Copying a string iii. Concatenating two

strings iv. Comparing

two strings

[Hint: Dialog boxes can be used wherever needed]

The Answer:

The code: package

pkg16bce0789_3;

```
/**
 *
 * @author OM MISHRA
 */

//package frames; import
javafx.geometry.Insets; import
javafx.application.Application; import
javafx.event.ActionEvent; import
javafx.event.EventHandler; import
javafx.geometry.Pos; import
javafx.scene.Scene; import
javafx.scene.control.Button; import
javafx.scene.control.Label; import
javafx.scene.control.TextField; import
javafx.scene.layout.GridPane; import
javafx.scene.text.Text; import
javafx.stage.Stage; import
javafx.application.Application; import
javafx.scene.Scene; import
javafx.scene.control.RadioButton;
import
javafx.scene.control.ToggleGroup;
import javafx.scene.layout.HBox;
import javafx.stage.Stage; package
```

```
frames; import java.util.Optional;
import javafx.application.Application;
import javafx.application.Platform;
import javafx.scene.Scene; import
javafx.scene.control.Alert; import
javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Menu;
import javafx.scene.control.MenuBar;
import javafx.scene.control.MenuItem;
import
javafx.scene.control.SeparatorMenuIte
m; import
javafx.scene.control.TextInputDialog;
import javafx.scene.layout.BorderPane;
import javafx.scene.paint.Color; import
javafx.stage.Stage; public class Main
extends Application{     void onadd()
   {
       //Reading first input for addition
TextInputDialog ip1 = new TextInputDialog();
ip1.setHeaderText("Give your input....");
ip1.setContentText("Enter an integer ");
       Optional<String> val1 = ip1.showAndWait();
       //Reading second input for addition
TextInputDialog ip2 = new TextInputDialog();
ip2.setHeaderText("Give your input....");
ip2.setContentText("Enter an integer ");
       Optional<String> val2 = ip2.showAndWait();
       //Converting string to int
int i1=Integer.parseInt(val1.get());
int i2=Integer.parseInt(val2.get());
```

```
        //Displaying sum
        Alert alert = new Alert(AlertType.INFORMATION);
       alert.setHeaderText("The result is...");
       alert.setContentText("Sum of the inputs is "+(i1+i2));
       alert.showAndWait();
    }

    void onsub()
    {
        //Reading first input for difference
TextInputDialog ip1 = new TextInputDialog();
ip1.setHeaderText("Give your input....");
ip1.setContentText("Enter an integer ");
        Optional<String> val1 = ip1.showAndWait();
        //Reading second input for difference
TextInputDialog ip2 = new TextInputDialog();
ip2.setHeaderText("Give your input....");
ip2.setContentText("Enter an integer ");
        Optional<String> val2 = ip2.showAndWait();
        //Converting string to int
int i1=Integer.parseInt(val1.get());
int i2=Integer.parseInt(val2.get());
        //Displaying diff
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("Difference of the inputs is "+(i1-i2));
alert.showAndWait();
    }

    void onmul()
    {
```

```
        //Reading first input for multiplication
        TextInputDialog ip1 = new TextInputDialog();
        ip1.setHeaderText("Give your input....");
        ip1.setContentText("Enter an integer ");

        Optional<String> val1 = ip1.showAndWait();
        //Reading second input for multiplication
TextInputDialog ip2 = new TextInputDialog();
ip2.setHeaderText("Give your input....");
ip2.setContentText("Enter an integer ");
        Optional<String> val2 = ip2.showAndWait();
        //Converting string to int
int i1=Integer.parseInt(val1.get());
int i2=Integer.parseInt(val2.get());
        //Displaying mul
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("Multiplication of the inputs is "+(i1*i2));
alert.showAndWait();
    }

    void ondiv()
    {
        //Reading first input for division
TextInputDialog ip1 = new TextInputDialog();
ip1.setHeaderText("Give your input....");
ip1.setContentText("Enter an integer ");
        Optional<String> val1 = ip1.showAndWait();
        //Reading second input for division
TextInputDialog ip2 = new TextInputDialog();
ip2.setHeaderText("Give your input....");
        ip2.setContentText("Enter an integer ");
```

```
        Optional<String> val2 = ip2.showAndWait();
        //Converting string to int int
        i1=Integer.parseInt(val1.get());
        int
        i2=Integer.parseInt(val2.get());
        //Displaying division
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("Division of the inputs is "+(i1/i2));
alert.showAndWait();
    }

    void oncopy()
    {
        //Reading the string to copy
        TextInputDialog ip1 = new TextInputDialog();
ip1.setHeaderText("Give your String");
ip1.setContentText("Enter an String ");
        Optional<String> val1 = ip1.showAndWait();
        String p = val1.get();
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("String copied is "+ p);
alert.showAndWait();
    }

    void onlen()
    {
        //Reading the string to copy
        TextInputDialog ip1 = new TextInputDialog();
```

```
        ip1.setHeaderText("Give your String");
       ip1.setContentText("Enter an String ");
       Optional<String> val1 = ip1.showAndWait();
       String p = val1.get();
        int k = p.length();
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("String length is "+ k);
alert.showAndWait();
    }

     void onconcat()
    {
        //Reading first string
        TextInputDialog ip1 = new TextInputDialog();
ip1.setHeaderText("Give your String");
ip1.setContentText("Enter an String ");
        Optional<String> val1 = ip1.showAndWait();
        //Reading second string
        TextInputDialog ip2 = new TextInputDialog();
ip2.setHeaderText("Give your String");
ip2.setContentText("Enter an String ");
        Optional<String> val2 = ip2.showAndWait();
        //Converting string to int
        String i1=val1.get();
        String i2=val2.get();
        String s = i1+i2;
        //Displaying concatenation
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("After Concatenation "+ s);
alert.showAndWait();
```

```
    }


     void oncompare()
    {
        //Reading first string
        TextInputDialog ip1 = new TextInputDialog();
ip1.setHeaderText("Give your String");
ip1.setContentText("Enter an String ");
        Optional<String> val1 = ip1.showAndWait();
        //Reading second string
        TextInputDialog ip2 = new TextInputDialog();
ip2.setHeaderText("Give your String");
ip2.setContentText("Enter an String ");
        Optional<String> val2 = ip2.showAndWait();
        //Converting string to int
        String
i1=val1.get();
String i2=val2.get();
int a = i1.length();
int b = i2.length();
        String c = a>b? "First String is longer" : "Second String is
Longer";
        //Displaying concatenation
        Alert alert = new Alert(AlertType.INFORMATION);
alert.setHeaderText("The result is...");
alert.setContentText("After Comaparing the length "+ c);
alert.showAndWait();
    }


   @Override    public void start(Stage
primaryStage) {
```

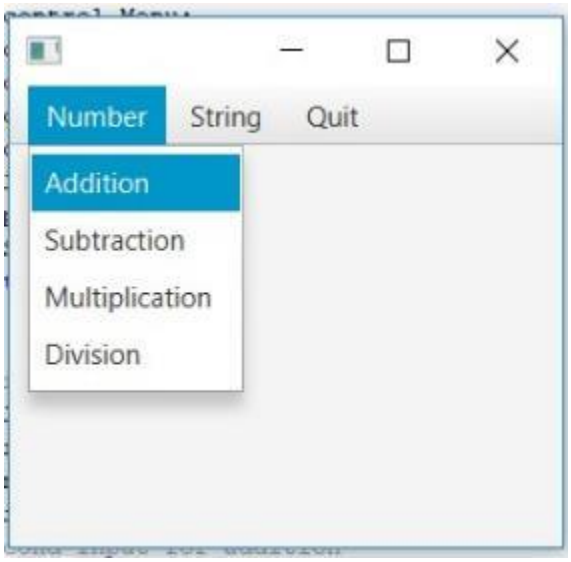
```
    MenuBar menubar = new MenuBar();
    Menu number = new Menu("Number");
MenuItem add = new MenuItem("Addition");
add.setOnAction(actionEvent -> this.onadd());
MenuItem sub = new MenuItem("Subtraction");
sub.setOnAction(actionEvent -> this.onsub());
MenuItem mul = new MenuItem("Multiplication");
mul.setOnAction(actionEvent -> this.onmul());
MenuItem div = new MenuItem("Division");
div.setOnAction(actionEvent -> this.ondiv());
number.getItems().addAll(add,sub,mul,div);

    Menu str = new Menu("String");      MenuItem copy
= new MenuItem("Copy");
copy.setOnAction(actionEvent -> this.oncopy());
MenuItem len = new MenuItem("Length");
len.setOnAction(actionEvent -> this.onlen());
MenuItem concat = new MenuItem("Concatenate");
concat.setOnAction(actionEvent -> this.onconcat());
MenuItem compare = new MenuItem("Compare");
compare.setOnAction(actionEvent -> this.oncompare());
str.getItems().addAll(copy,len,concat,compare);

    Menu quit = new Menu("Quit");
    MenuItem exit = new MenuItem("Exit from the application");
exit.setOnAction(actionEvent -> Platform.exit());
quit.getItems().add(exit);

    menubar.getMenus().addAll(number,str,quit);
BorderPane bp = new BorderPane();
```

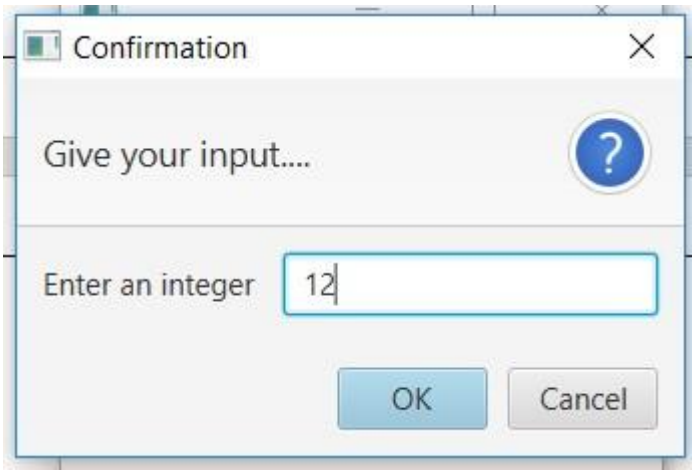
```
      Scene scene = new Scene(bp, 300, 250,
Color.WHITE);       bp.setTop(menubar);
primaryStage.setScene(scene);
primaryStage.show();
   }
   public static void main(String[] args) {
launch(args);
  }
}
```

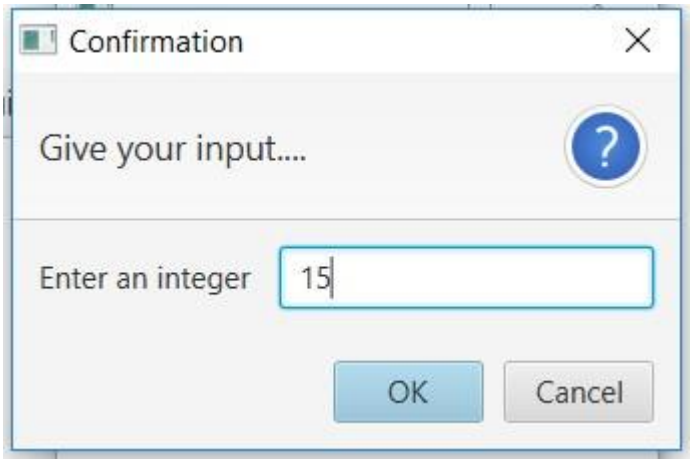
The Output:

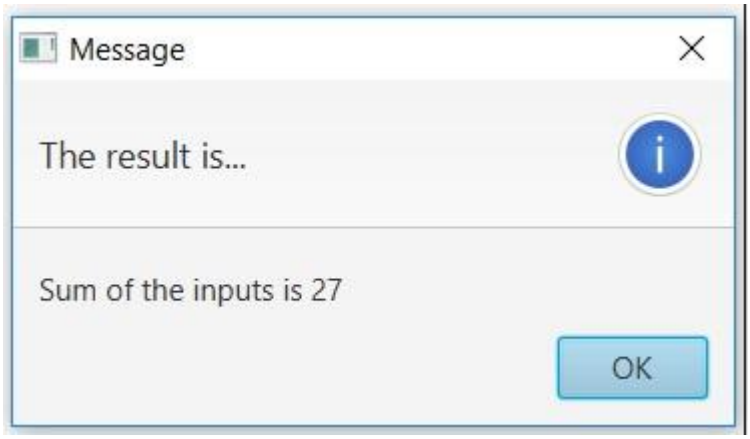
For Numbers

The addition



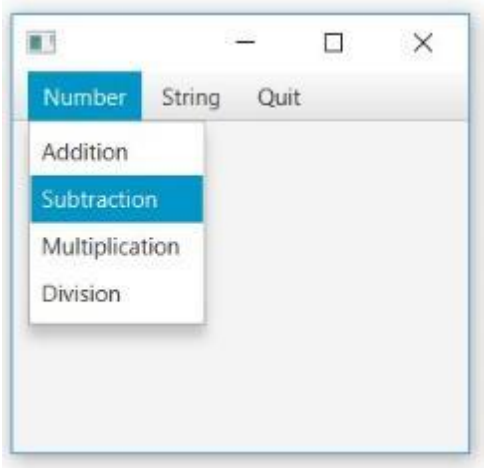


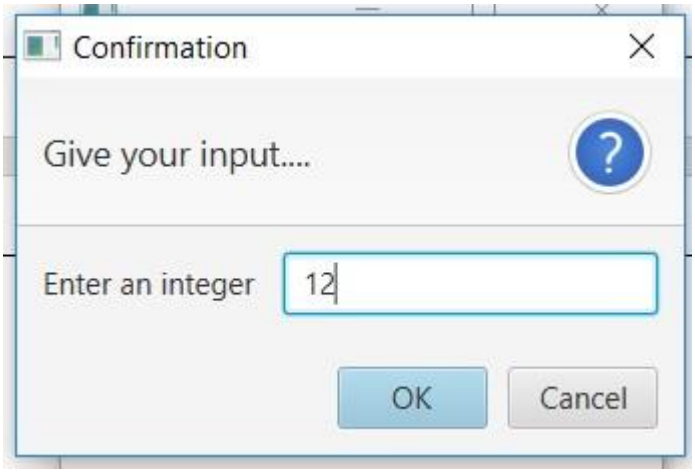


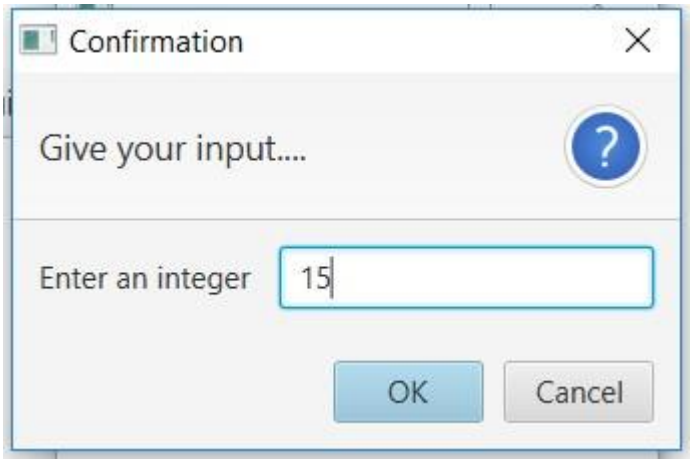


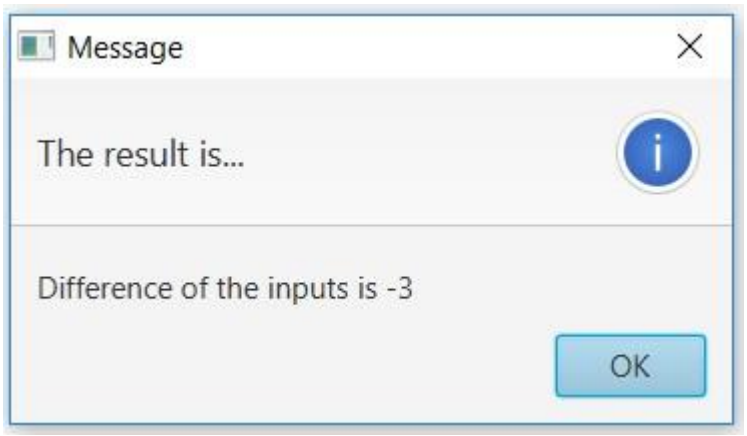
Similarly for Subtraction, Multiplication and Division.

The Subtraction:

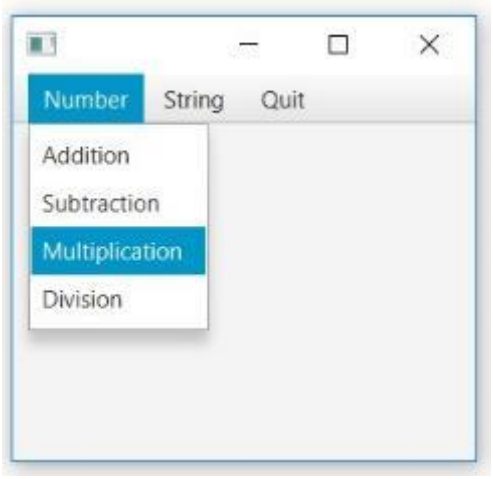


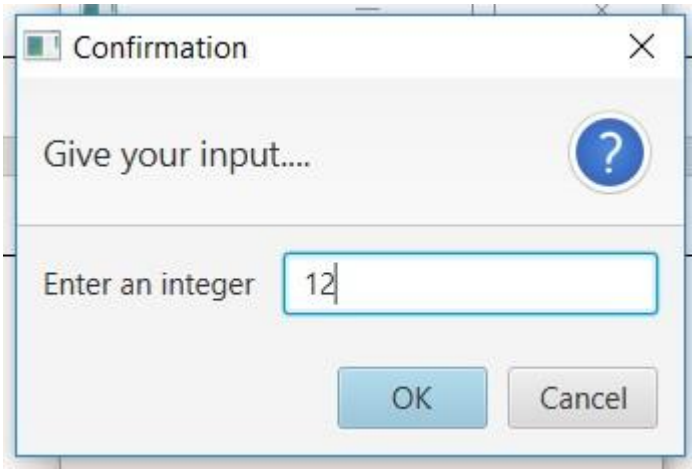


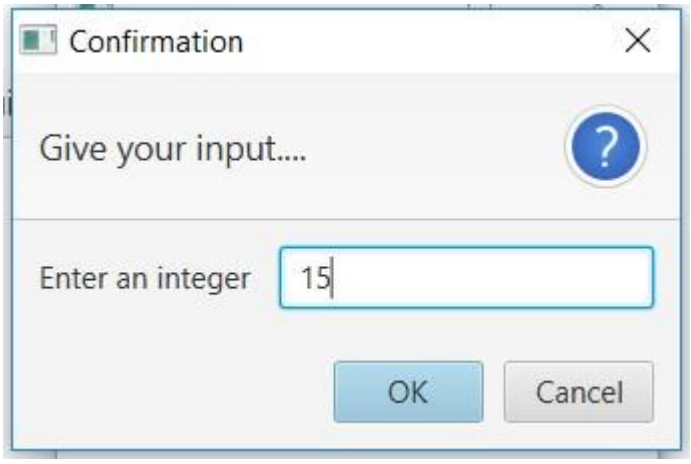


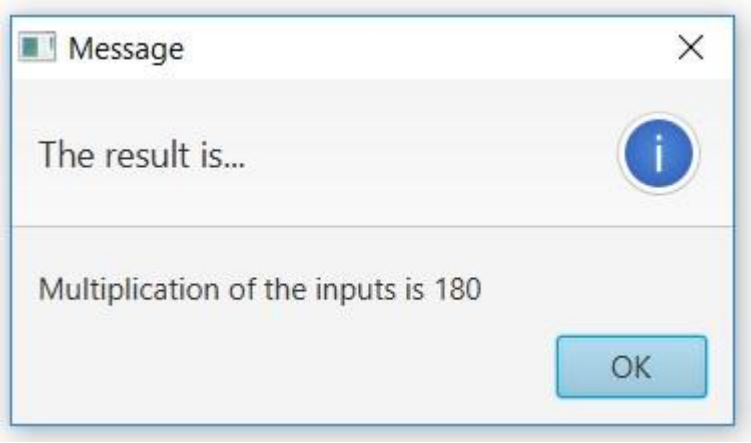


The Multiplication

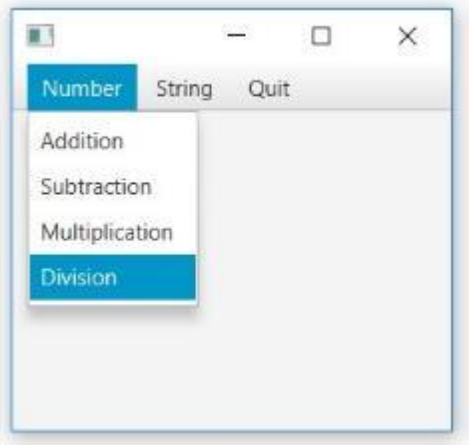

Number
String
Quit
Addition
Subtraction
Multiplication
Division

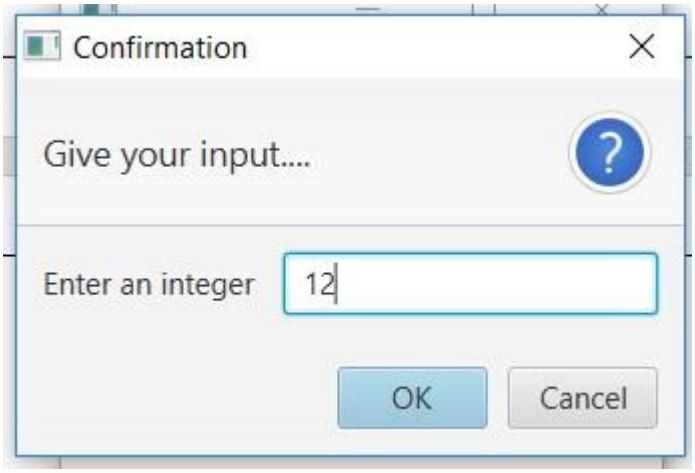

Confirmation
Give your input....
Enter an integer
12
OK
Cancel

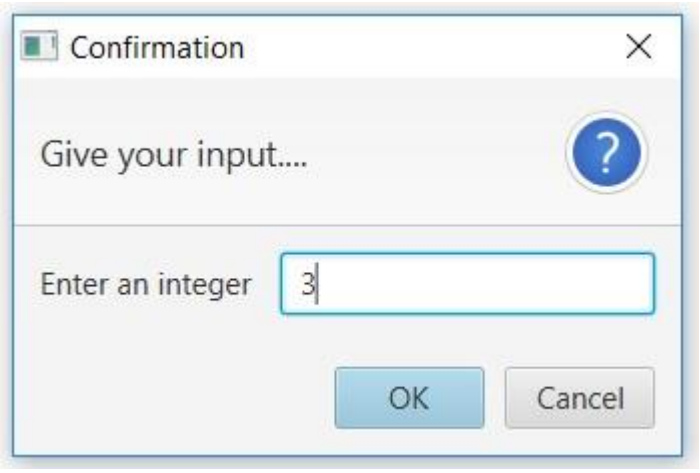

Confirmation
Give your input....
Enter an integer
15
OK
Cancel

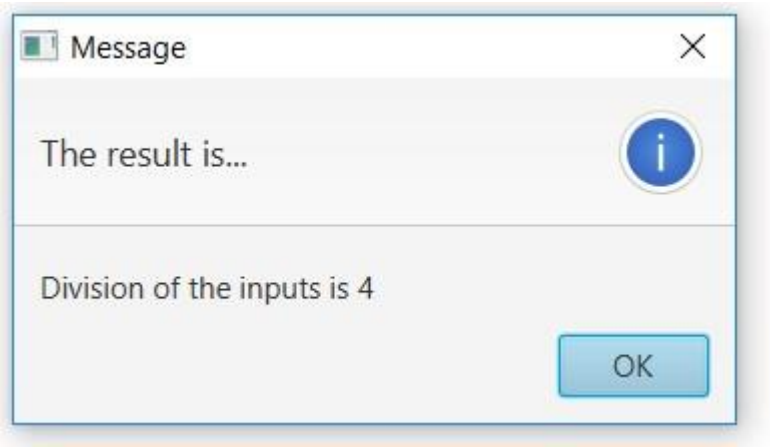


The Division



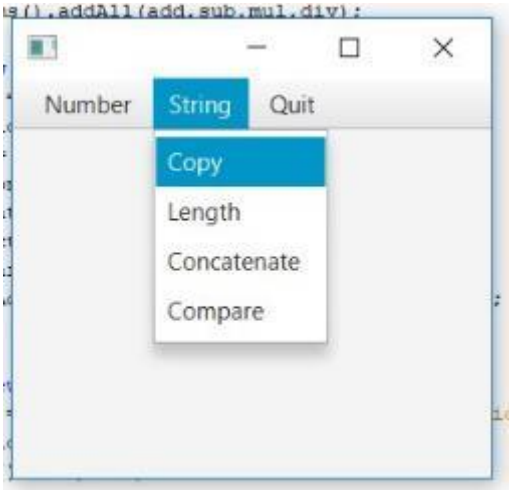




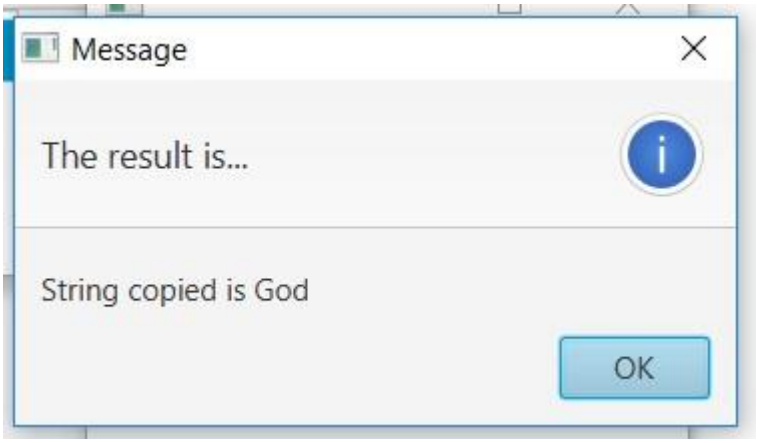


For Strings

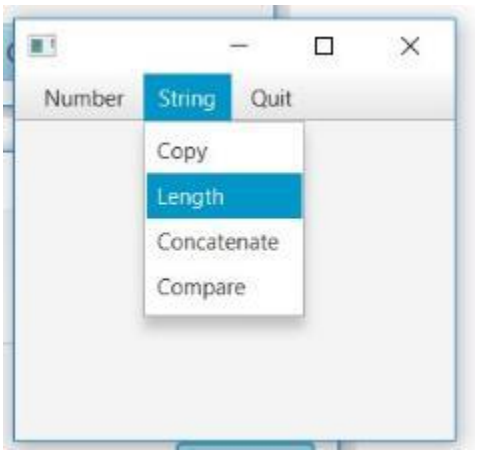
The Copying

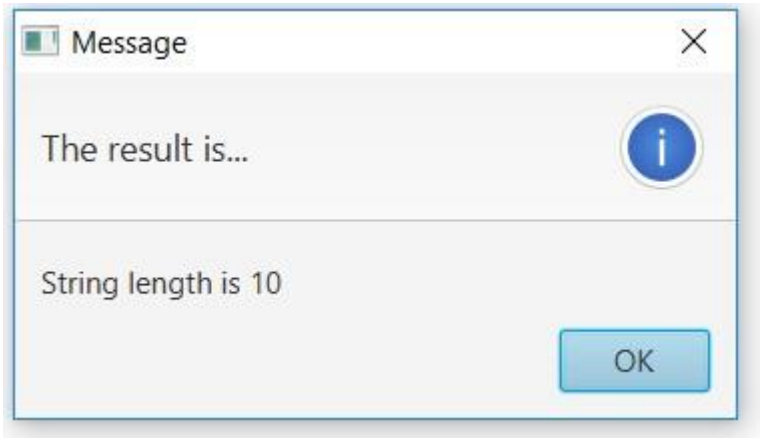



Confirmation
Give your String
Enter an String
God
OK
Cancel

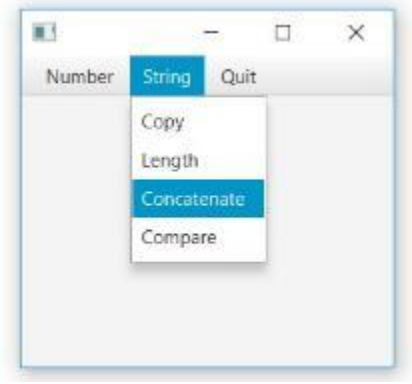

Message
The result is...
String copied is God
OK

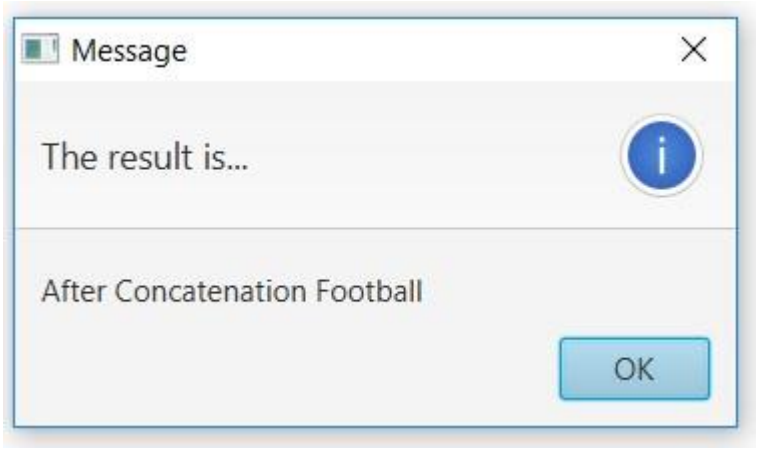
The Length


Number
String
Quit
Copy
Length
Concatenate
Compare

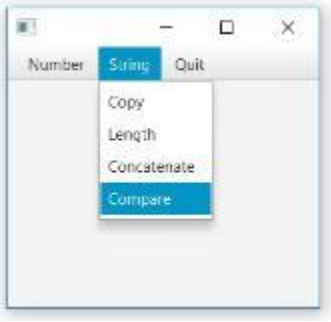


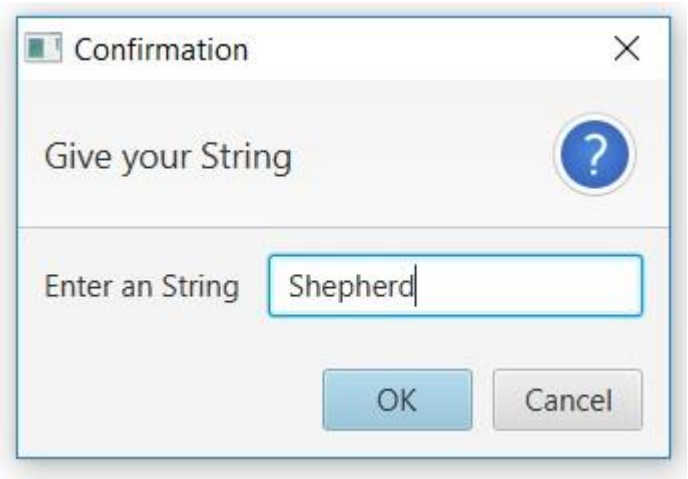
The Concatenation

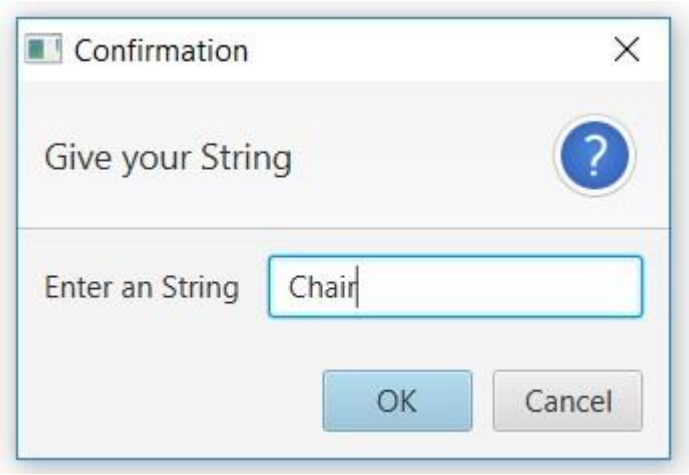


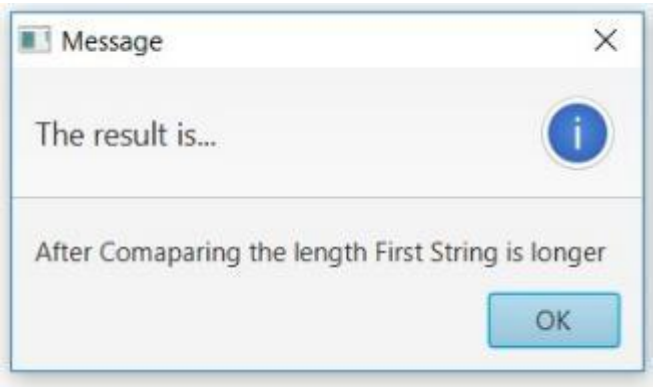


The Comparing

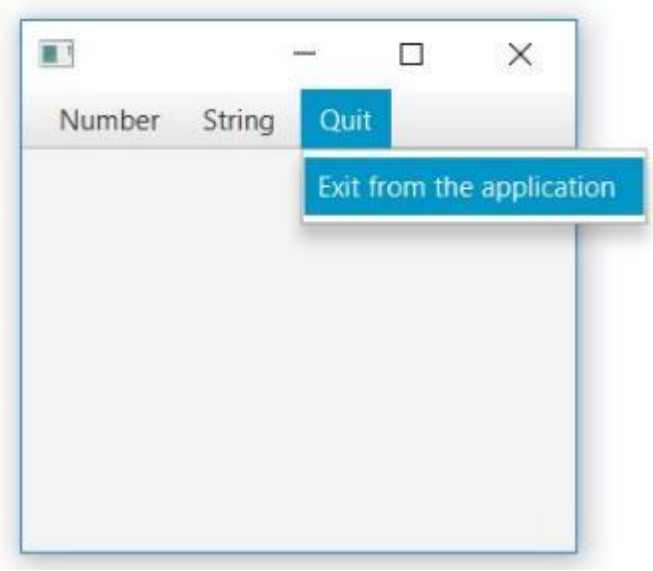








The Quitting



Exception handling Program 1:

```
class JavaException
{  public static void main(String args[])
 {   int d = 10;   int n = 20;   int
fraction = 0;   int g[] = {1};   try {
fraction = n / d;    g[20] = 100;
   System.out.println("This line will not be Executed");
    }
 /*catch(Exception e){
            System.out.println("In the catch clock due to Exception = "+e);
 }*/
 catch (ArithmeticException e) {
System.out.println("In the catch clock due to Exception = " + e);
   }
 catch (ArrayIndexOutOfBoundsException e) {
System.out.println("In the catch clock due to Exception = " + e);
  }   finally{
                    System.out.println("Inside the finally block");
            }

 System.out.println("fraction = "+fraction);
 System.out.println("End Of Main");
 }
}
```

Program 2:

```
import java.io.*; class MyException extends Exception {
private double amount;              public MyException(double
amount) {           this.amount = amount;
            }
            public double getAmount() {                 return
amount;
            }
          }
class CheckingAccount {
            private double balance;           private int
number;

            public CheckingAccount(int number) {            this.number =
number;
            }

   public void deposit(double amount) {       balance += amount;
            }

            public void withdraw(double amount) throws MyException {
if(amount <= balance) {                 balance -= amount;              }else {             double
needs = amount - balance;               throw new MyException(needs);
              }
            }

            public double getBalance() {                return
balance;
            }

            public int getNumber() {           return number;
            }
          }
public class BankDemo {

              public static void main(String [] args) {
               CheckingAccount c = new CheckingAccount(101);
System.out.println("Depositing $500...");           c.deposit(500.00);
try {
                 System.out.println("Withdrawing $100...");              c.withdraw(100.00);
                 System.out.println("Withdrawing $600...");              c.withdraw(600.00);
               } catch (MyException e) {
                 System.out.println("Sorry, but you are short $" + e.getAmount());
e.printStackTrace();
              }
            }
          }
```

Inheritance

Hierarchical inheritance Program 1:

```
class A1 {    public void methodA()
  {
    System.out.println("method of Class A");
  } } class B extends A1
{   public void methodB()
  {
    System.out.println("method of Class B");
  } } class C extends A1
{
 public void methodC()
 {
   System.out.println("method of Class C");
 } } class D extends A1
{   public void methodD()
 {
   System.out.println("method of Class D");
 } } public class hierarchical {

          public static void main(String args[])
          {
            B obj1 = new B();       C obj2 =
new C();              D obj3 = new D();
obj1.methodA();              obj2.methodA();
   obj3.methodA();
          }

}
```

Multi-level inheritance Program 2:

```
class X {    public void methodX()
  {
   System.out.println("Class X method");
  } } class Y extends X
{ public void methodY()
{
System.out.println("class Y method");
} } class Z extends Y
{   public void methodZ()
  {
   System.out.println("class Z method");
  }
  public static void main(String args[])
  {
```

```
    Z obj = new Z();      obj.methodX();
obj.methodY();       obj.methodZ();
  }
}
```

Program 3:

```
class poly
{
        poly()
        {
                System.out.print("Hello");
        }
        void area()
        {
        }
} class sqr extends poly
{
        protected int side; sqr(int s)
        {
                super();
//this();                       this.side=s;
                }
        void area()
        {
                 System.out.println("The area is "+(side*side));
        }
} class cube extends sqr
{
        cube(int a)
        {
                super(a);
        }
        void area()
        {
                 System.out.println("The area is "+6*(side*side));
        }
} public class polygon {


         public static void main(String[] args) {
                // TODO Auto-generated method stub      poly obj;
obj=new sqr(10);      obj.area();      obj=new cube(5);      obj.area();
        }

}
```

Inner classes Program 4:

```
class outer
{
        class inner
        {
                int a;
                inner(){a=0;}           void
print()
                {
                        System.out.print(a+++" ");
                }
        }
        inner obj;      int i;
outer()
        {
                obj=new inner();
        }
   void display(int n)
   {
        for(i=0;i<n;i++)
                obj.print();
   }
} public class innerclass {     public static void
main(String[] args) {
                outer ob=new outer();           ob.display(10);
                 System.out.println();
                 for(int i=0; i<args.length; i++)
                        System.out.println("args[" + i + "]: " + args[i]);

        }

}
```

Polymorphism Program 5:

```
   class Bank{     float getRateOfInterest(){return
0;}
   }
   class SBI extends Bank{     float
getRateOfInterest(){return 8.4f;}
   }
   class ICICI extends Bank{     float
getRateOfInterest(){return 7.3f;}
   }
```

```
class AXIS extends Bank{ float
getRateOfInterest(){return 9.7f;}
}
class TestPolymorphism{ public static void
main(String args[]){ Bank b; b=new SBI();
System.out.println("SBI Rate of Interest: "+b.getRateOfInterest()); b=new ICICI();
System.out.println("ICICI Rate of Interest: "+b.getRateOfInterest()); b=new AXIS();
System.out.println("AXIS Rate of Interest: "+b.getRateOfInterest());
}
}
```

Abstract class Program 6:

```
abstract class Person {
        private String name;  private String desig;
public Person(String nm, String des){
                this.name=nm;
this.desig=des;
        }
        //public abstract void work();  public String toString(){                return "Name is
"+this.name+" and the designation is "+this.desig;
        }
        public void changeName(String newName) {          this.name =
newName;
        }
} public class Employee extends Person {      private int empId;
public Employee(String nm, String des, int id) {
                super(nm, des);
this.empId=id;
        }
        public void work() {
                if(empId == 0){
                        System.out.println("Not working");
                }else{
                        System.out.println("Working as employee!!");
                }
        }
        public static void main(String args[]){
                Employee student = new Employee("Shakthi","Team Lead",0);
                  Employee employee = new Employee("Karthik","Developer",123);
                /*Person student = new Employee("Shakthi","Team Lead",0);
                Person employee = new Employee("Karthik","Developer",123);
Person p=new Person("aaaa","bbbb",12);*/              student.work();
employee.work();
                employee.changeName("Goutham");
```

```
                System.out.println(employee.toString());
        }

}
```

final - keyword usage Program 7:

```
final  class c1
{
        final
        void display()
        {
                System.out.print("Hello in c1");
        }
} class c2 //extends c1
{
        void display()
        {
                System.out.print("Hello in c2");
        }
} public class final_class {                public static void
main(String[] args) {    c2 obj = new c2();
obj.display();
        }
}
```

Interfaces Program 8:

```
interface inf1 {     int x=10; } interface inf2 {
int x=100; } class Hello implements inf1,inf2
{
   public static void main(String args[])
   {
     /* reference to x is ambiguous both variables are x
* so we are using interface name to resolve the
* variable
      */
     //System.out.println(x);
     System.out.println(inf1.x);
     System.out.println(inf2.x);
   }
}

public class Example{
public static void main(String args[]){
String s1="hello";
String s2="whatsup";
```

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

String s3="hello";
String s4="hello";
String s5="hemlo";
String s6="flag";
System.out.println(s3.compareTo(s4)); // 0 because both are equal
System.out.println(s3.compareTo(s5)); //-1 because "l" is only one time lower than "m"
System.out.println(s3.compareTo(s6)); // 2 because "h" is 2 times greater than "f"

String s7="hello";
s7=s7.concat("how are you");
System.out.println(s7);

String s8="";
String s9="hello";
System.out.println(s8.isEmpty());      // true
System.out.println(s9.isEmpty());      // false

String s10="  hello   ";
System.out.println(s10+"how are you");        // without trim()
System.out.println(s10.trim()+"how are you"); // with trim()

String s11="HELLO HOW Are You?";
String s11lower=s11.toLowerCase();
System.out.println(s11lower);

String s12="hello how are you";
String s12upper=s12.toUpperCase();
System.out.println(s12upper);

int value=20;
String s13=String.valueOf(value);
System.out.println(s13+17);       //concatenating string with 10

String s14="hello how are you";
String replaceString=s14.replace('h','t');
System.out.println(replaceString);

String s15="Hey, welcome to Edureka";
String replaceStr=s15.replace("Edureka","Brainforce");
System.out.println(replaceStr);

String name=" hello how are you doing?";
System.out.println(name.contains("how are you"));  // returns true
System.out.println(name.contains("hello"));        // returns true
```

```
System.out.println(name.contains("fine"));        // returns false

String s16="hello";
String s17="hello";
String s18="hi";
System.out.println(s16.equalsIgnoreCase(s17));   // returns true
System.out.println(s16.equalsIgnoreCase(s18));   // returns false

String s19="hello";
String s20="HELLO";
String s21="hi";
System.out.println(s19.equalsIgnoreCase(s20));   // returns true
System.out.println(s19.equalsIgnoreCase(s21));   // returns false

String s22="Welcome to Edureka";
char[] ch=s22.toCharArray(); for(int
i=0;i<ch.length;i++){
System.out.print(ch[i]);
}

String s23="ABC"; byte[]
b=s23.getBytes();        for(int
i=0;i<b.length;i++){
System.out.println(b[i]);
}

String s24="";
String s25="hello";
System.out.println(s24.isEmpty());     // returns true
System.out.println(s25.isEmpty());     // returns false

String s26="hello how are you";
System.out.println(s26.endsWith("u"));       // returns true
System.out.println(s26.endsWith("you"));     // returns true
System.out.println(s26.endsWith("how"));     // returns false
}}
//String vs StringBuffer....Performance Test of String and StringBuffer     public
class ConcatTest{
      public static String concatWithString()      {
String t = "Java";                for (int i=0; i<10000;
i++){
            t = t + "Tpoint";
         }
         return t;
      }
      public static String concatWithStringBuffer(){
StringBuffer sb = new StringBuffer("Java");
```

```
        for (int i=0; i<10000; i++){
            sb.append("Tpoint");
        }
        return sb.toString();
    }
    public static void main(String[] args){
long startTime = System.currentTimeMillis();
        concatWithString();
        System.out.println("Time taken by Concating with String:
"+(System.currentTimeMillis()-startTime)+"ms");
startTime = System.currentTimeMillis();
concatWithStringBuffer();
        System.out.println("Time taken by Concating with  StringBuffer:
"+(System.currentTimeMillis()-startTime)+"ms");
    }
  }

//Java Program to demonstrate the performance of StringBuffer and StringBuilder classes.   public
class ConcatTest{
  public static void main(String[] args){          long
startTime = System.currentTimeMillis();
StringBuffer sb = new StringBuffer("Java");
    for (int i=0; i<10000; i++){
        sb.append("Tpoint");
    }
    System.out.println("Time taken by StringBuffer: " + (System.currentTimeMillis() -
startTime) + "ms");
    startTime = System.currentTimeMillis();
StringBuilder sb2 = new StringBuilder("Java");
    for (int i=0; i<10000; i++){
        sb2.append("Tpoint");
    }
    System.out.println("Time taken by StringBuilder: " + (System.currentTimeMillis() -
startTime) + "ms");
  }
}


// Java program to demonstrate difference between String,
// StringBuilder and StringBuffer
class Geeksforgeeks
{
  // Concatenates to String
  public static void concat1(String s1)
  {
    s1 = s1 + "forgeeks";
  }
```

```
    // Concatenates to StringBuilder
    public static void concat2(StringBuilder s2)
    {
        s2.append("forgeeks");
    }

    // Concatenates to StringBuffer
    public static void concat3(StringBuffer s3)
    {
        s3.append("forgeeks");
    }

    public static void main(String[] args)
    {
        String s1 = "Geeks";
concat1(s1);  // s1 is not changed
        System.out.println("String: " + s1);

        StringBuilder s2 = new StringBuilder("Geeks");
concat2(s2); // s2 is changed
        System.out.println("StringBuilder: " + s2);

        StringBuffer s3 = new StringBuffer("Geeks");
concat3(s3); // s3 is changed
        System.out.println("StringBuffer: " + s3);
    }
}


// Java program to demonstrate conversion from   //
String to StringBuffer and StringBuilder.  public class
Test
{
    public static void main(String[] args)
    {
        String str = "Geeks";

        // conversion from String object to StringBuffer
StringBuffer sbr = new StringBuffer(str);
        sbr.reverse();
        System.out.println(sbr);

        // conversion from String object to StringBuilder
StringBuilder sbl = new StringBuilder(str);
        sbl.append("ForGeeks");
        System.out.println(sbl);
```

```
    }
}


// Java program to demonstrate conversion from   //
String to StringBuffer and StringBuilder.  public class
Test
{
    public static void main(String[] args)
    {
        StringBuffer sbr = new StringBuffer("Geeks");
        StringBuilder sbdr = new StringBuilder("Hello");

        // conversion from StringBuffer object to String
        String str = sbr.toString();
        System.out.println("StringBuffer object to String : ");
        System.out.println(str);

        // conversion from StringBuilder object to String
        String str1 = sbdr.toString();
        System.out.println("StringBuilder object to String : ");
        System.out.println(str1);

        // changing StringBuffer object sbr          //
but String object(str) doesn't change
sbr.append("ForGeeks");
System.out.println(sbr);
        System.out.println(str);

    }
}
```