JAVA PROGRAMMING

LAB MANUAL

Faculty of Science & Technology
The ICFAI University
Dehradun

INDEX

Assignment -1 Java Basics

Commands for unix environment	
mkdir hari cd hari	To make directory To change directory
vi t.java i Esc :w :q	To enter into editor To enter into insert mode To leave insert mode To save To quit from editor
javac t.java java mainclass	To compile file t.cpp To execute

```
class Triangle
{
   public static void main(String args[])
   {
    int height =10, base=6;
    float area=0.5F*base* height;
    System.out.println("area of triangle = "+area);
  }
}
```

Program prints area of a triangle.

```
class Commd
{
    public static void main(String args[])
    {
        System.out.println(args[0]+" "+args[2]);
    }
}
```

Program takes input from command line.
For e.g. **java Commd** Fst Icfai University
And program prints **Fst University**

```
class Cs
{
    public static void main(String args[])
    {
       int x=Integer.parseInt(args[0]);
       int k=Integer.parseInt(args[1]);
       for(int i=0; i<k; i++)
       {
             x=x/10;
       }
       System.out.println("no. after deleting last
       "+k+ " digits = "+x);
     }
}
```

Program deletes last k digits of a number x; where input is taken from command line arguments. (x is args[0] and k is args[1])

```
class Cs
{
   public static void main(String args[])
   {
       int x=Integer.parseInt(args[0]);
       int d,y=0,t=1;
       while (x>0)
       {  d=x%10;
          if (d%2==0)
          {  y=y+t*d;
             t=t*10;
          }
          x=x/10;
        }
        System.out.println("Number after
        deleting odd digits = "+y);
  }
```

Program deletes odd digits.

1. Write program to print the k^{th} digit from last. e.g. input 23617 and k=4 output 3.
2. Write a program to print first digit. e.g. input 23516 output 2.
3. Write program to print the second digit. e.g. input 23516 the output is 3.
4. Write program to find sum of all digits. Input 23617 output 2+3+6+1+7=19.
5. Write program, which will find sum of product to consecutive digits. e.g. when the input is 23145 the output is 2x3 + 3x1 + 1x4 + 4x5 = 33.
6. Write program, which reads two number (assume that both have same number of digits). The program outputs the sum of product of corresponding digits. Input 327 and 539 output 3x5+2x3+7x9=84.
7. Write program to print positional values of digits. Input 21463 output 3, 60, 400, 1000 and 20000.
8. Write program to find sum of even digits. Input 23617 output 2+6=8.
9. Write program to find number of digits. Input 423 output 3. Input 21151 output 5.

10. Write program to find number of even digits. In above case 2 and 1 respectively.
11. Write program to print the last even digit. e.g. input 23613 output 6.
    [Hint:while (x%2 $\neq$ 0) x=x/10]
12. Program to print the digit immediately before the last even digit. In above case 3.
13. Write program to print the digit immediately after the last even digit. In above case 1.
14. Write program to print the last digit, which is multiple of 3. e.g. input 23617 output 6.
15. Write program to print the second last even digit. e.g. input 23863 output 8 (do not use ‘if’). Input 325145761 output 4. [Hint: use two loops]
16. Read a number. Do half of number after last odd digit. Input 3 times. Input 61389426 output 184167639 (61389213*3). Input 87 output 261. Input 78 output 222 (74*3).
17. Write program, which finds the sum of numbers formed by consecutive digits. Input 2415 output 24+41+15=80.
18. Find sum of numbers formed by exchanging consecutive digits. In above 42+14+51=107.

Assignment -2 Classes & Objects

```
class Student
{
   String name, city;
   int age;
   static int m;
   void printData()
   {
      System.out.println(“Student name
      = “+name);
      System.out.println(“Student city
      = “+city);
      System.out.println(“Student age
      = “+age);
   }
}

class Cs
{
 public static void main(String args[])
 {
   Student s1=new Student();
   Student s2=new Student();
   s1.name=”Amit”;
   s1.city=”Dehradun”;
   s1.age=22;
   s2.name=”Kapil”;
   s2.city=”Delhi”;
   s2.age=23;
   s2.printData();
   s1.printData();
   s1.m=20;
   s2.m=22;
   Student.m=27;
   System.out.println(“s1.m = “+s1.m);
   System.out.println(“s2.m = “+s2.m);
   System.out.println(“Student.m =
   “+Student.m);
 }
}
```

Program shows basic functionality of java class,object, and static variable;

```
class Student
{
   private String name, city;
   private int age;
   public void getData(String x, String
   y, int t)
   {
      name=x;
      city=y;
      age=t;
   }
   public void printData()
   {
      System.out.println(“Student name =
      “+name);
      System.out.println(“Student city =
      “+city);
      System.out.println(“Student age =
     “+age);
   }
}
class Cs
{
  public static void main(String args[])
  {
      Student s1=new Student();
      Student s2=new Student();
      s2.getData(“Kapil”,”Delhi”,23);
      s2.printData();
      s1.getData(“Amit”,”Dehradun”,22);
      s1.printData();
  }
}
```

```
class Point
{ private float x,y;
  public void getPoint(float a,float b)
  {x=a;y=b;}
  public void print()
  {
   System.out.println("("+x+","+y+")");
  }
  public void abc() {x=2*x;y=3*y;}
  public static Point pqr(Point a)
  { Point t;
    t=new Point();
    t.getPoint(2*a.x,2*a.y);
    return t;
  }
  public void ghi(Point a)
  {  x=2*a.x;y=3*a.y;  }
  public float getx()
  {  return x;   }
}

-------------->     ------------->
```

```
class hari
{
 public static void main(String args[])
 {   Point a,b,c;float t;
     a=new Point();
     a.getPoint(2,4);
     a.print();
     a.abc();
     a.print();
     b=Point.pqr(a);
     b.print();
     a=new Point();
     a.getPoint(2,4);
     c=new Point();
     c.ghi(a);
     c.print();
     t=a.getx();System.out.println(t);
  }
}
```

In above if point p is (x,y) then, p.abc( ) will make it (2x,3y). If point r is (x,y) then, q=point.pqr will make q as (2x,2y). If point k is (x,y) then, r.ghi(k) will make r as (2x,3y)

1. Modify first example for taking input through command line arguments.
2. Modify second example for taking input through command line arguments.
3. Define fa. If point p is (x,y) then p.fa() will make it (x+y,2y). (20,4)⟶(24,8)
4. Define fb. If point p is (x,y) then p.fb() will make it (2x,x+y). (20,4)⟶(40,24)
5. Define fc. If point p is (x,y) then p.fc() will make it (x+y,x*y). (20,4) ⟶(24,80)
6. Define fd. If point p is (x,y) then q=p.fd() will make q as (x+y,x*y). [local variable]
7. Define ga. If point p is (x,y) then q=p.ga() will make q as (x+y,2y).
8. Define gb. If point p is (x,y) then q=point.gb(p) will make q as (x+y,2y).
9. Define gb using ga. In its definition + and * should not be used.
10. Define gc. If point p is (x,y) then q.gc(p) will make q as (x+y,2y).
11. Define gc using ga. In its definition + and * should not be used.
12. Define gd. If point p is (x,y) then p.gd(k) will make it (x+k,2y).
13. Define ga, gb and gc using gd. In their definition + and * should not be used.

Assignment -3 Method Overloading & Constructors

```
class Cs
{
  int p,q;
  public Cs(){}
  public Cs(int x, int y)
 {
     p=x;
     q=y;
  }
  public int add(int i, int j)
  {
     return (i+j);
  }

  public int add(int i, int j, int k)
  {
     return (i+j+k);
  }

  public float add(float f1, float f2)
  {
     return (f1+f2);
  }
  public void printData()
  {
    System.out.print(“p = “+p);
    System.out.println(“ q = “+q);
  }
}

class Hari
{
 public static void main(String args[])
 {
    int x=2, y=3, z=4;
    Cs c=new Cs();
    Cs c1=new Cs(x, z );
    c1.printData();
    float m=7.2F, n=5.2F;
    int k=c.add(x,y);
    int t=c.add(x,y,z);
    float ft=c.add(m, n);
    System.out.println(“k = “+k);
    System.out.println(“t = “+t);
    System.out.println(“ft = “+ft);
 }
}
```

Program explains the concept of method overloading and constructor overloading.

```
import java.lang.*;
class Comp
{ float real,img;
  public void getComp(float a,float b)
   { real=a;img=b; }
   public void print1()
   {
   System.out.println(real+"+"+img+"i");
   }
   public void duble()
   { real=2*real; img=2*img; }
   public void ktimes(int k)
   {  real=k*real;img=k*img; }
   private float magsq()
   { return real*real+img*img;}
   public float magnitude()
   { float t;
     t=(float)Math.sqrt(this.magsq());
     return(t);
   }
   public void add(Comp x)
   { real=real+x.real;img=img+x.img; }
   public void add(Comp x,Comp y)
   {
     real=x.real+y.real;img=x.img+y.img;
   }
   public Comp add3(Comp x)
   { Comp c;float a,b;
     c=new Comp();
     a=real+x.real;
     b=img+x.img;
     c.getComp(a,b);
     return(c);
   }
   public float getreal()
   { return(real); }
   private float getimaginary()
   { return(img);}
}
class hari
{public static void main( String args[])
 { Comp t,s,r,m;float j;
   t=new Comp();s=new Comp();
   r=new Comp();
   t.getComp(4,2);t.duble();t.print1();
   t.getComp(4,2);t.ktimes(10);
   t.print1();
   t.getComp(4,3);j=t.magnitude();
   System.out.println(j);
   s.getComp(2,7);t.getComp(4,2);
   s.add(t);s.print1();
   s.getComp(2,7);t.getComp(4,3);
   r.add(s,t);r.print1();
   s.getComp(2,7);t.getComp(4,23);
   r=s.add3(t);r.print1();
   System.out.println(s.getreal());
 }
}
```

Since getreal( ) is public s.getreal( ) can be written in class hari. However s.getimaginary( ) can not be written.

s.add(t) means s+=t; r.add(s,t) means r=s+t; Two definitions show method overloading.

r=s.add3(t) means r=s+t;

1. Write a JAVA program which contains a method square() such that square(3) returns 9, square(0.2) returns 0.04.
2. Write a JAVA program which contains a method cube() such that cube(3) returns 27, cube(0.2) returns 0.008.
3. Write a JAVA program which contains a method fun() such that fun(x) returns x^2 and fun(x,y) returns $x^2 + y^2$. (where x and y are integers).
4. Write a JAVA program which contains a method fun() such that fun(x) returns x and fun(x,y) returns x + y and fun(x,y,z) returns x*y*z. (where x, y and z are integers).
5. Write a set of overloaded methods **min()** that returns the smaller of two numbers passed to them as arguments. Make versions for int and float.
6. Write a set of overloaded methods **power()** that returns the X^n where n is integer and X may be int and float.
7. Write a set of overloaded methods **max()** that returns the biggest of two numbers passed to them as arguments. Make versions for int and float.
8. Implement a=a*b as a.mul(b), where a and b are objects of class Comp.
9. Implement a=b*c as a=Comp.mul(b, c), where a, b and c are objects of class Comp.
10. Define Comp conjugate() e.g. conjugate(2+3i) is 2-3i
11. Implement a=b/c as a=b.divide(c), where a, b and c are objects of class Comp.
    `[Hint: a/b is defined as a*b.conjugate()/b.magnitude`^{2}`()]`
12. Implement a=b/c as a=Comp.divide(b, c), where a, b and c are objects of class Comp.
    `[Hint: a/b is defined as a*b.conjugate()/b.magnitude`^{2}`()]`

Assignment -4 Array, Searching & Sorting

```
class Avg
{
public static void main(String
                          args[])
  {
      int n=args.length;
      float [] x=new float[n];
      for(int i=0; i<n; i++)
      {
        x[i]=Float.parseFloat(
                        args[i]);
      }
      float sum=0;
      for(int i=0; i<n; i++)
          sum=sum+x[i];
      float avg=sum/n;
      System.out.println("Average of
             given numbers is
             "+avg);
   }
}
```

Program finds average of numbers in an array.

```
class Add
{
 public static void main(String
                           args[])
 {
  int [][] x={{1,2,3},
                {4,5,6},
                {7,8,9}
               };
  int [][] y={
                 {11,12,13},
                 {14,15,16},
                 {17,18,19}
                };
  int [][] z=new int[3][3];
  for(int i=0; i<3; i++)
     for(int j=0; j<3; j++)
     {
         z[i][j]=x[i][j]+y[i][j];
     }

  for(int i=0; i<3; i++)
  {
    for(int j=0; j<3; j++)
    {
      System.out.print(z[i][j]+” ”);
    }
    System.out.print(“\n”);
  }
 }
}
```

Program finds addition of two matrices.

1. Find smallest number in an array.
2. Find largest number in an array.
3. Count even numbers in an array.
4. Count occurrence of a given number in an array.
5. Check if given number is palindrome or not.
6. Input two arrays and merge them in a new array in ascending order.
7. Find Addition of two 3X3 matrices.
8. Find Multiplication of two 3X3 matrices.
9. Find Transpose of a given matrices.
10. Implement Binary Search.
11. Implement Bubble Sort.
12. Implement Selection Sort.
13. Implement Insertion Sort.

Assignment -5 String

In a file name (say a.java) type following program. The file can be compiled by command **javac a.java**. To run the program give command **java kapil**.

```
import java.io.*;
import java.lang.*;
class kapil
{  public static void main(String args[])
   { String a,b;
     try
     { DataInputStream o=new DataInputStream(System.in);
       a=o.readLine();
       b=a.substring(2,5);
       System.out.println(b);
     }
     catch(IOException e) { System.out.println(e);  }
   }
}
```

The program outputs sub string between positions 2 and 5 (including 2 but excluding 5). The first character is at 0th position. e.g. input qwertyuiuo output ert.

```
       b=a.substring(4);System.out.println(b);
```

The program outputs string on and after 4th position. input qwertyuiuo output tyuiuo

```
        String a,b,c; a=o.readLine();b=o.readLine();
       c=a+b; System.out.println(c);
```

The program takes two strings and joins them.

1. Write program to remove 2nd letter. Let the input string is pwsxtpbcderxrtxgt then output is pwxtpbcderxrtxgt.
2. Write program to add 0th letter in the beginning. In above case ppwsxtpbcderxrtxgt
3. Write program to exchange first two letters. In above case wpsxtpbcderxrtxgt.
4. Write program to exchange 4th and 10th letter. In above case pwsxrpbcdetxrtxgt.
5. Write program to insert ‘t’ between 1st and 2nd letter. In above case pwtsxtpbcderxrtxgt. [Hint: a+”t”+b].

```
       a=o.readLine();
       i=a.indexOf('x');
       System.out.println(i);
```

At what location 'x' is present. If more than one occurrence of 'x' is there then the location of first 'x' is returned. If 'x' is absent then –1 is returned. e.g. input wedxtyhxu output 3

```
       int i;String a,b,c,d;i=a.indexOf('x');
       b=a.substring(0,i);c=a.substring(i+1);
       d=b+c; System.out.println(d);
```

The first 'x' in the given string is deleted.

6. Write a program, which reads a string and finds string after the first x. Let the input string is pwsxtpbcderxrtxgt then output is tpbcderxrtxgt.
7. Write program to replace first x by y. In above case pwsytpbcderxrtxgt.
8. Write program to output the location of second x. In above case 11.
9. Write program to print the string between 1st and 2nd x. In above case tpbcder.
10. Write program to find string before 2nd x. In above case pwsxtpbcder.
11. Write program to delete the string between 1st and 2nd x. In above case pwsxxrtxgt.
12. Program to exchange the string between 1st and 2nd x, with the string before 1st x. In above case tpbcderxpwsxrtxgt
13. Write program to exchange neighbors of first x. In above case pwtxspbcderxrtxgt.

```
char b;a=o.readLine();b=a.charAt(2);
System.out.println(b);
```

Program outputs character at location 2. e.g. input qwertyuiuo output e

14. Write program to find 2nd location of 0th letter. Input pwerpty output 4.
15. Write program, which will delete 1st y immediately after 1st x. If input string is pgyeryuyixaysdyexer then output is pgyeryuyixasdyexer.
16. Write program to exchange neighbors of first occurrence of left neighbors of first ‘x’. e.g. input abcdefxgh output abcdxfegh. input abfcdefxgh output acfbdefxgh.
17. Write program to replace first occurrence of right neighbor of 2nd x by left neighbor of 1st x. Input imgpxugxutkl output imgpxpgxutkl. Input bcxdefxgh output bcxdefxch.
18. Write program, which reads a string. Let x and y be respectively left and right neighbors of the second occurrence of the 0th letter. Find the substring between first occurrence of y and (first occurrence of x after first occurrence of y). e.g. input patkgfmpkst output kgfm. Input pastgksfsptse output tgks. Input raklfrgmcfd output gmcf. Input ywetyykjhtl output ywet.

```
a=o.readLine();b=o.readLine();
i=a.compareTo(b);System.out.println(i);
```

Input two strings. Output is 0 if both are same. If second string is (lexicographically) bigger then a negative number is outputted. If first string is bigger then some positive number is outputted.

Use of “if” is permitted in following programs.

19. Read two strings. Print lexicographically bigger string first and smaller later.
20. Read two string. Print 1 if first string is bigger, 2 if second string is bigger, 0 if both are same.
21. Read three strings. Print 1 if first string is biggest, 2 if second string is biggest, 3 if 3rd string is biggest, 0 if all are same, -1 if 1st and 2nd string are biggest, -2 if 2nd and 3rd string are biggest, -3 if 1st and 3rd string are biggest.

```
a=o.readLine();a=a.trim();
i=a.indexOf(' ');b=a.substring(0,i);
System.out.println(b);
```

Print first word. Trim removes blank spaces at the beginning and at the end. If it is not used then the problem will arise if blanks are given at the beginning. The program will not work if string has only one word. If string is ram Prasad dey then output is ram.

22. Print second word. In above case Prasad.
23. Delete second word. In above case ram dey.
24. Exchange first and second word. In above case Prasad ram dey.
25. Exchange first letters of first two words. In above case Pam rrasad dey.
26. Exchange last letters of first two words. In above case rad Prasam dey.
27. Find the location of first ‘a’ in second word. In above case 6.
28. Find location of first letter of first word in second word. In above case 5.

Assignment -6 String Buffer

```
import java.io.*;
import java.lang.*;
class kapil
{  public static void main( String args[])
   { StringBuffer a;String b;int i;
     try
     { DataInputStream o=new DataInputStream(System.in);
       a=new StringBuffer("");b=o.readLine();a.append(b);
       System.out.println("The input string is"+a);
       i=a.indexOf("x");System.out.println(i);
       i=a.indexOf("xy");System.out.println(i);
       a.setCharAt(6,'X');System.out.println(a);
       a.insert(4,"tty");System.out.println(a);
       a.append("ghij");System.out.println(a);
       a.reverse();System.out.println(a);
       a.delete(4,7);System.out.println(a);
       a.deleteCharAt(5);System.out.println(a);
       a.replace(3,7,"was");System.out.println(a);
       a.setLength(5);System.out.println(a);
       b=a.toString();System.out.println(b);
     }
     catch(Exception e)
     { System.out.println("Error"+e);  }
   }
}
```

All the programs should be done without using +(plus), –(minus) or append (except during converting input string to string buffer).

1. Read a string and replace first x by y. Input artxuxtxxp output artyuxtxxp.
2. Replace last x by y. (do not use loop)
3. Replace second x by y.
4. Replace all x's by y.
5. Find the number of x's. In above case 4.
6. Whether given string is a Palindrome. A Palindrome is word which remains same when reversed. e.g. malayalam.
7. Find length of the string. (Do not use *length( )*)
8. Suppose string has words. e.g. Ram is a good boy. Find first and second word.
9. Find the first word of 4 letters. In above case *good*. Assume such word present at least once.
10. Remove all blanks. In above case *Ramisagoodboy*.
11. Find last letter of first word.
12. Find first letter of last word.
13. Find first letter of every word.
14. Replace first letter of every word by x. In above case x*am xs x xood xoy*.
15. Word wise reverse of the string. In above case *boy good a is Ram*.
16. Reverse each word. In above case output is *maR si a doog yob*.
17. Interchange first two words. In above case *is Ram a good boy*.
18. Write program, which reads an expression and outputs its value. Assume that parenthesis is not used. Assume all operations are left associative and all numbers are integers. Take only '+' and '*' as operators. e.g. input 12+17*10 output 290.

Assignment - 7 Arithmetic

```
int k;String a;
     try
     { DataInputStream o=new DataInputStream(System.in);
       a=o.readLine();k=Integer.parseInt(a);
       if (k%2==0) System.out.println(a+" is even");
       else System.out.println(a+" is odd");
     }
```

Program reads a number and finds whether it is even or odd. The program does not work if blanks are left in the beginning or at the end. Hence a=a.trim() should be used. It removes blanks at the beginning and at the end.

```
     int i;float x,y,k,l;String a,b;
     try
     { DataInputStream o=new DataInputStream(System.in);
       a=o.readLine();a=a.trim();
       i=a.indexOf(" ");b=a.substring(0,i);x=Float.parseFloat(b);
       b=a.substring(i+1);y=Float.parseFloat(b.trim());
       k=x+y;l=x*y;System.out.println(k+" "+l);
     }
```

Program reads two numbers and outputs their sum and product. Both numbers are given in same line.

1. Read a string of digits. Let n be its first digit. Find the number formed by next n digits. Output double of the number. Input 4773962 output 7739*2=15478.
2. Write program, which reads two numbers. It prints the bigger number first and smaller later.
3. Write program, which reads a complex number and outputs its modulus. Input 4+3i output 5. [Hint: p=(float)Math.sqrt(q) will find square root of q]
4. Write program, which reads a vector and outputs its magnitude. Input 3i+4j+12k output 13.
5. Read two vectors and print their dot product. Input 5i+8j+0k and 6i–3j+2k output 6.
6. Write program, which behaves in the following manner: Input 12+10 output 22. Input 12-10 output 2.
7. Read a string. It contains a word and two numbers. The word is one of the followings: add, sub, mul, and div. The program outputs the sum of given numbers, if the word is add. If word is sub the program outputs difference and so on. e.g. input mul 12 19 output 228.
8. Write program, which reads a string. Its first number is either 1 or 2. If it is 1 then square of second number is returned. If it is 2 then product of 2nd and 3rd number is returned. Input 1 12 output 144. Input 2 5 12 output 60.
9. Read a float number and output its double. Do not use parseFloat. Assume fraction part has exactly two digits. Input 12.34 output 24.68. Input 862.97 output 1725.94.
10. Read a string. It has 2 or 3 numbers. If it has two numbers then output their product. If it has three numbers then output their sum.
11. Write a program, which reads a number *n* and prints a string with *n* x’s. There will be only one print statement. Input 6 output xxxxxx.
12. Read a string of numbers and find 5th number. Input 6 2 6 4 17 8 9 18 output 17.
13. Read a string of 5 numbers and find their sum. Input 6 403 2 5 7 output 423.
14. Read a string of numbers and find first even number. Input 61 3 6 43 17 8 9 18 output 6.
15. Read a string of numbers and find their sum. Input 6 3 2 5 7 output 23. Input 2 5 3 output 10. The number of numbers in the string is unknown.
16. Write program, which deletes all even numbers from the string. Input 12 17 19 6 17 output 17 19 17. [Caution: all numbers should come in same line]

17. Read a complex number and print it in simplified form. Input 7i+12i+9+4i-8i+13 output 22+15i.
18. Read a string. The first word is a number (say k). The program finds k^{th} word in remaining string. e.g. input 3 Ram Hari Gopal Om Ravi output Gopal.
19. Write program, which reads an expression and finds its value. Input (((6+3)*8)–(4*3)) output 60.
20. Write a program, which reads number n and n strings. The program outputs the string, which has the longest length.
21. Write program, which reads a complex number and outputs its modulus. Input 4+3i output 5. Input 4 output 4. Input 3i output 3. Input 4-3i output 5.
22. Read two complex numbers and find their sum. Input 5-7i and 3 output 8-7i.
23. Write program, which will delete maximum number from the string. Assume all numbers are distinct. Try to do it using only one for loop.
24. Write program for deleting maximum number, when numbers are not distinct. If maximum number occurs more than once then first occurrence is deleted. Input 12 17 6 13 17 6 17 9 output 12 6 13 17 6 17 9.
25. Write program, which will delete all occurrences of maximum number.
26. The program reads two strings of numbers. In both the numbers are in increasing order. The program outputs all numbers in increasing order. e.g. if input strings are 6 9 14 20 22 25 and 11 12 15 16 21 then output is 6 9 11 12 14 15 16 20 21 22 25.
27. Do above problem to find k^{th} word from last. Use only one for.
28. Read a string of numbers. Let these are x_0, x_1, x_2, … , x_{n-1}. Without using array find the value of $(x_0 - (x_1 - (x_2$ ……….. $- (x_{n-4} - (x_{n-3} - (x_{n-2} - x_{n-1}))) \ldots)))$

Assignment - 8 Recursion

```
public static int loc(String a,char e,int k)
   {  String b;int i,j;
      if (k==1) return a.indexOf(e);
      else{ i=a.indexOf(e);b=a.substring(i+1);
            j=loc(b,e,k-1);return(j+i+1); }
   }
   public static String del(String a,int k)
   {   String b,c,d;int i,j;
       if (k==0)  return a;
       else{  i=a.indexOf('x');b=a.substring(i+1);c=a.substring(0,i);
              d=del(b,k-1);return(c+d); }
   }
class kapil
{  public static void main( String args[])
   { String a,k;int g;
     a=f.rdstring();g=Integer.parseInt(f.rdstring());
     System.out.println(f.loc(a,'x',g));
     System.out.println(f.del(a,g));
   }
}
```

The method loc finds the location of kth e in string a. The method del deletes first k x's.

Define following methods without using any loop.

1. Define method int count(String a, char e). It returns the number of e’s in the string ‘a’. count(“wrtkcrygm”,’r’) will return 2.
2. Define a method String delfirst(String a). It deletes first ‘x’ in string x. Using it define method to delete (A) first k x’s. (B) all x’s. In them indexOf should not be used.
3. Define method to return short form. short(“Ram Prasad Kumar Dey”) returns “R.P.K.Dey”. short(“shyam singh”) returns s.singh.
4. Define method, which returns the string of those words, which has at least one ‘x’. abc(“ram haxi pxxy wertm xrt”) will return “haxi pxxy xrt”.
5. Define method, which returns the string of second letter of those words whose first letter is ‘x’. ghi(“xram mhaxi tpxxy xxwertm xgrt”) will return “rxg”.
6. Modify the method “loc” (given in example). It returns -1 if string does not have k e’s.
7. Define method String subset(String a, String b). It returns “yes” if every letter of first string is present in second string. It returns “no” otherwise.
8. Define method String present(String a, String b). It returns string of those words of string ‘a’ whose every letter is present in ‘b’. present(“ram is a good boy”,”tsoaigkdm”) will return “isagood”. [Hint: use subset].
9. Define method common. It returns “yes” both strings have got some common letter. common(“hari”,”ram”) returns “yes”. common(“hari”,”som”) returns “no”.
10. Define method to remove duplicates. rd(“hariram”) returns “harim” (or “hiram”).
11. Define method, which returns the location of last ‘x’. If the string does has any ‘x’ then -1 is returned.
12. Define method to delete the last ‘x’. Assume that string has at least one ‘x’.
13. Define method to delete second last ‘x’. Assume that string has at least two x’s.
14. Define method to find the location of kth last x. If the string does not have k x’s then -1 is returned. kthlast(“ramxttxharixxkapilxravi”,4) will return 6.
15. Define method to find the location of kth last x from last. If the string does not have k x’s then -1 is returned. kthlast(“ramxttxharixxkapilxravi”,4) will return 17.
16. Define method to find the number of letters after last ‘x’. Assume that string has at least one ‘x’. [Hint: To find the length of string a.length() may be used.

Assignment -9 Exception Handling

```
import java.io.*;
import java.lang.*;
class xyz
{ public static void main (String args[])
  { String a,b;char c; a=”abc”;
    try{ b=a.substring(2,5); System.out.println(b);
         c=a.charAt(9);System.out.println(c);
       } catch (Exception t){ System.out.println(“The exception is”+t); }
    System.out.println(“Hari”);
  }
}
```

The above program outputs “Hari” irrespective of input string ‘a’. When input is abc or abcdef then the exception is java.lang.StringIndexOutOfBounds. In case of “abc” no additional output is produced. In case of “abcdef” output “cde” is produced. When input is abcedfghijkl then there is no exception. When try-catch is removed then program terminates when input is abc or abcedf. In case of “abc” no output is produced. In case of “abcdef” output “cde” is produced. Because of try-catch the impact of exception is felt only within the block. The normal execution is carried out after that.

```
class xyz
{
  public static void main (String args[])
  {   int a,b,c,d,e,f;a=25;b=4;c=0;d=3;
      e=a/b;System.out.println(e);
      e=a/c;System.out.println(e);
      e=a/d;System.out.println(e);
      System.out.println(“Hari”);
  }
}
```

The above program terminates after outputting 6.

```
Class xyz
{
  public static void main (String args[])
  {   int a,b,c,d,e,f;a=25;b=4;c=0;d=3;
      try{
        e=a/b;System.out.println(e);
        e=a/c;System.out.println(e);
        e=a/d;System.out.println(e);
       }
      catch (ArithmeticException t){System.out.println(“mistake”);}
      System.out.println(“Hari”);
  }
}
```

The above program outputs 6 mistake hari

```
Class xyz
{
  public static void main (String args[])
  {   int a,b,c,d,e,f;
      a=25;b=4;c=0;d=3;
      try{e=a/b;System.out.println(e);}
         catch (ArithmeticException t){System.out.println(“Error”);}
      try{e=a/c;System.out.println(e);}
         catch (ArithmeticException t){System.out.println(“Error”);}
      try{e=a/d;System.out.println(e);}
         catch (ArithmeticException t){System.out.println(“Error”);}
  }
}
```

The above program outputs 6 error and 8.

```
Class xyz
{ private static int read1()
  { int i=2,j;String a;
    try{
        DataInputStream o=new DataInputStream(System.in);
        a=o.readLine();i=Integer.parseInt(a);}
    catch (Exception t){ System.out.println(“Not a number”+t); }
    return i;
  }
  public static void main (String args[])
  {   int i,a,f[]={8,12,34,16};
      a=xyz.read1();
      try
      {  System.out.println(f[a]);
         i=25/0;
      }
      catch (Exception t){ System.out.println(“stop”+t);}
      /*catch (ArrayIndexOutOfBoundsException t)
          {System.out.println(“Illegal Memory reference”+t);}
      catch (ArithmeticException t)
          {System.out.println(“Wrong Division”+t);}*/
  }
}
```

input 2 Output 34 stop java.lang.ArithmeticException: / by zero

input 7 Output stop java.lang.ArrayIndexOutOfBoundsException: 7

input hari Output Not a number java.lang.NumberFormatException
34 stop java.lang.ArithmeticException: / by zero
For this input initialize i as 3 and 5 in method read1.

When catch exception is removed and other two catches are activated then for input 2 the output is “wrong division”. For input 7 the output is “Illegal Memory Reference”.

When all three catches remain then program refuses to compile. It is because catch exception suppresses other two exceptions. However, when catch exception is written at the last, the program compiles. When catch arithmetic exception and catch exception is written then for input 2 the output is wrong division. For input 7 the output is “Stop”.

```
Class xyz
{
  public static void main (String args[])
  {   int a,b,c,d,e,f; a=25;b=4;c=0;d=3;
      try{
        e=a/b;System.out.println(e);
        e=a/c;System.out.println(e);
        e=a/d;System.out.println(e);
     }
     finally{System.out.println(“Continue”);}
     e=a/d;System.out.println(e);
  }
}
```

The above program outputs 6 Continue.

When c=0 is replaced by c=10 then output is 6 2 8 Continue 8.

When “finally” is replaced by “catch(Exception t)” then output is 6 Continue 8 when c is 0.

When “finally” is replaced by “catch(Exception t)” then output is 6 2 8 8 when c is 10. “Finally” means in case of exception the program will execute the corresponding statement. The program will terminate after that. In case of “catch exception” the program continues after that. When exception does not occur then the statement in finally is executed. But statement is catch exception is not executed.

```
class kapil
{  public static void main( String args[])
   { int i;float x,y,k,l;String a=””,b;
     try
     { DataInputStream o=new DataInputStream(System.in);
       a=o.readLine();a=a.trim();
     }catch(IOException e){ }
     i=a.indexOf(“ “);b=a.substring(0,i);
     try{x=Float.parseFloat(b);}catch(Exception e){x=5;}
     b=a.substring(i+1);y=Float.parseFloat(b.trim());
     k=x+y;l=x*y;System.out.println(k+” “+l);
   }
}
```

Program reads two numbers and outputs their sum and product. If first number is wrong then it is taken as 5.

```
       k=Integer.parseInt(a);k=k%2;
       try{p=”ram”;p=p.substring(k+3);
           System.out.println(a+” is an even number”);
       }catch(Exception m){System.out.println(a+” is an odd number”);}
```

Program reads a number and finds whether it is even or odd.

1. Read a string. If it is a number then output its double otherwise output 0. Input 12 output 24. Input 3.9 output 7.8. Input “12 14” output 0. Input ram output 0.
2. Read a number. If it a integer then output its double. If it is a float then output its triple. Do not use indexOf or if.
3. Read a string. If it is a number then output “AC”. Otherwise output “BC”. You should use only 3 print statements. Every print statement should print only one letter.
4. Modify above program: in case of number output ABC. Otherwise output “B”. [Hint: use finally]
5. Read a string. It has 1 or 2 numbers. If it has one number the output its square. If it has two numbers then output their sum. Do not use if.
6. Read a string. It has 2 or 3 numbers. If it has two numbers then output their product. If it has three numbers then output their sum. Do not use if.
7. Read a string of two (distinct) integers. Output bigger integer. Do not use “if”.
8. Read a string of two (distinct) numbers (floats). Output bigger number. Do not use “if”.
9. Read a string of two numbers. Output 1 if first number is bigger. Output 2 if second number is bigger. Output 0 if they are equal. Do not use “if”.
10. Read a string. If it is a number then output 0. Otherwise let ‘p’ be the 0th letter. If p is not an integer then output 1. If p is an integer and size of the string is less than p+1 then output 2. Otherwise if string becomes integer after deleting pth letter then output 3. Otherwise output 4. 157 → 0 r23er → 1 5123a → 2 332a67 → 3 2r7 → 4 33abc45 → 4.
11. Read a number and output its double. Do not use parseFloat. Assume fraction part has at most one digits. Input 12.3 output 24.6. Input 218 output 436. Do not use if.
12. Read a string of two numbers (words) and output their sum. If first word is not a number then it is taken as 5. If second word is not a number then it is taken as 7. Input 12 34 output 46. Input ram 30 output 35. However if both are not numbers then output is 100. Do not use if.
13. Write program, to find number of integers in a string. Input rat 12 3.0 34 om output 2.
14. Write program, which reads a string of digits and finds how many digits are more than 5. Input 2145835769 output 3. Do not use if. Use only ArrayIndexOutOfBoundsException.

15. Write program, which reads a string of digits and finds how many digits are equal to 5. Input 2145835769 output 2. Do not use if. Use only ArrayIndexOutOfBoundsException.
16. Write program, which reads a string. It finds how many of them are integers and how many of them are floats. E.g. Input “2 ram 4 10.3 20 4.78 20 3” output 5,7.

Write following program without using length method, “if” and any loop except while(1==1).

17. Write program, which reads a string and find its length.
18. Write program, which finds number of words in a string.
19. Given a string of integers, find program to find how many of them are more than 15.

Assignment - 10 Inheritance and Interface

Inheritance

```
import java.io.*;
import java.lang.*;

class d2vectors {
     float i,j;
     public d2vectors(float i1,float j1){
      i=i1;
      j=j1;
    }
    public float slope(){return (j/i); }
    public float magsqr(){return (i*i+j*j);}
    public void dble(){i=2*i;j=2*j;}
    public void pt(){System.out.println(i+"i+"+j+"j");}
    public float geti(){return (i);}
    public float getj(){return (j);}
}

class d3vectors extends d2vectors{
    float k;
    public d3vectors(float i1, float j1, float k1)
    {
      super(i1,j1);
      k=k1;
    }
    public float magsqr(){return(super.magsqr()+k*k);}
    public void ttt(){super.dble();k=k*3;} //Remove super
    public void dble(){i=2*i;j=2*j;k=2*k;}
    public double direction(){return (k/Math.sqrt(i*i+j*j));}
    // public d2vectors ttt( ){return super;}
    // super can not be used like keyword "this"
    public void pt(){System.out.println(i+"i+"+j+"j+"+k+"k");}
}

class xyz{
      public static void main(String ar[])
      {
        d2vectors a,c;double t;
        d3vectors b,d;
        b=new d3vectors(6,8,7);
        c=new d2vectors(6,7);
        t=c.magsqr();System.out.println(t);
        t=b.magsqr();System.out.println(t);
        t=b.slope();System.out.println(t);
        t=b.direction();System.out.println(t);
        c.pt();b.pt();
        b.dble();b.pt();
        b.ttt();b.pt();
      }
}
```

1. Define class stack { private String x[]; int sp; ….put take top print empty size …}
2. In same file Define class queue { private x[]; int sp; ….put take top print empty size …}
3. Write main program which reads a word. If it is "stack" then remaining operations are according to stack. If it is "queue" then remaining operations are according to queue.
4. Define class queue extends stack{ int front; … take top print empty size ….} Do not define "put"
5. Define class queue extends stack{ …. put….} Do not define other operations
6. Define class priority extends stack{ int front; ….Put….}

Interface

```
import java.io.*;
import java.lang.*;
interface picture {
   float area ();
}
class rectangle implements picture
{  float p,q;
   public rectangle(float a, float b){p=a;q=b;}
   public float area (){ return p*q;}
}
class circle implements picture
{  float r;
   public circle(float a){r=a;}
   public float area () {return (float)3.14*r*r;}
}
class xyz
{  public static void main( String args[])
    {  picture p;float a,b;int i;String s,t,u,v;
       try{
        DataInputStream o=new DataInputStream(System.in);
        s=o.readLine();s=s+" 10 ";
        i=s.indexOf(' ');
        t=s.substring(0,i);
        v=s.substring(i+1);
        i=v.indexOf(' ');
        u=v.substring(0,i);
        s=v.substring(i+1);
        i=s.indexOf(' ');
        s=s.substring(0,i);
        a=Float.parseFloat(u.trim());
        b=Float.parseFloat(s);
        if (t.compareTo("circle")==0)
           p=new circle(a);
        else
  //      if (t.compareTo("rectangle")==0)
            p=new rectangle(a,b);
        a=p.area();
        System.out.println("area="+a);
      }
      catch (IOException e){ System.out.println(e);}
   }
}
```

Give input circle 4 and rectangle 3 7

Assignment - 11 Multithreading

```
class xyz implements Runnable
{   public void run()
    {   int i;
        for (i=1;i<5;i++)
        {   System.out.print(i);
            try{Thread.sleep(1000);}catch(Exception e){}
        }
    }
}
class kapil
{
    public static void main(String ar[])
    {   xyz k;Thread a,b;
        k=new xyz();a=new Thread(k);b=new Thread(k);
        a.start();b.start();
        System.out.print("X");
    }
}
```

The above program outputs X1**1**2**2**3**3**4**4** (or 1X**1**2**2**3**3**4**4** or 1**1**X2**2**3**3**4**4)** When a.start( ) is replaced by a.run( ) then the output is 1234X**1234**. When b.start( ) is replaced by b.run( ) then the output is 1**1**2**2**3**3**4**4**X.

```
class xyz implements Runnable
{   int k;
    xyz(int g) {k=g;}
    public void run()
    {   int i;
        for (i=k;i<k*2;i++)
        {   System.out.print(i);
            try{Thread.sleep(1000);}catch(Exception e){}
        }
    }
}
class kapil
{
    public static void main(String ar[])
    {   xyz x,y;Thread a,b;
        x=new xyz(5);a=new Thread(x);
        y=new xyz(3);b=new Thread(y);
        a.start();b.start();
    }
}
```

The above program outputs 5**3**6**4**7**5**89.

```
class xyz implements Runnable
{   public void run()
    {   int i;
        for (i=1;i<6;i++)
        {   System.out.print(i);
            try{Thread.sleep(1000);}catch(Exception e){}
        }
    }
}
class pqr implements Runnable
{   public void run()
    {   xyz t;Thread c;
        System.out.print("X");
        try{Thread.sleep(3000);}catch(Exception e){}
        System.out.print("Y");
        t=new xyz();c=new Thread(t);c.start();
    }
}
class kapil
{
    public static void main(String ar[])
    {   xyz k;pqr g;Thread a,b;
        k=new xyz();a=new Thread(k);
        g=new pqr();b=new Thread(g);
        a.start();b.start();
    }
}
```

The above program outputs 1**X**23**Y1**4**2**5**345**

```
class xyz implements Runnable
{   int k,count=0;;
    xyz(int g) {k=g;}
    public void run()
    {   xyz h;Thread b,c;
        try{Thread.sleep(k*1000);}catch(Exception e){};
        if (k>1)
        {   h=new xyz(k-1);
            b=new Thread(h);c=new Thread(h);
            b.start();c.start();
        }
        count++;
        System.out.print("["+k+","+count+"]");
    }
}
class kapil
{
    public static void main(String ar[])
    {   xyz x;Thread a;
        x=new xyz(5);a=new Thread(x);
        a.start();
    }
}
```

```
class f implements Runnable
{  public void run()
   {  int i;
      for (i=1;i<10;i++)
      {   System.out.print(i);
          try {Thread.sleep(2000);}catch(Exception e){}
      }
   }
}
class g implements Runnable
{  public void run()
   {  int i;
      for (i=65;i<=73;i++)
      {   System.out.print((char)i);
          try {Thread.sleep(1000);}catch(Exception e){}
      }
   }
}
class kapil
{  public static void main(String ar[])
   {  f x;g y;Thread a,b;
      x=new f();a=new Thread(x);
      y=new g();b=new Thread(y);
      a.run();b.start();
      try {Thread.sleep(3000);}catch(Exception e){}
      System.out.print("ravi");
   }
}
```

The output is 123456789ABCraviDEFGHI

Assignment - 12 Abstract Data Type

```
class stack
{  private String x[]=new String[10];int sp;
   public stack(){int sp=0;}
   public void put(String e) {x[sp]=e;sp++;}
   public String take() {sp--; return x[sp];}
   public String top() {return x[sp-1];}
   public String empty()
                   {if (sp==0) return “empty”; else return “not empty”;}
   public int size() { return sp;}
   public void pt(){int I;for (i=0;i<sp;i++) System.out.print(x[i]+” “);}
}
class kapil
{  public static void main( String args[])
   { String a=””,b;int I;stack x;
     x=new stack();
     do{
       try
       { DataInputStream o=new DataInputStream(System.in);
         a=o.readLine();
       }
       catch(IOException e) {}
       i=a.indexOf(‘ ‘);
       if (i!=-1) b=a.substring(0,i);else b=a;
       if (b.compareTo(“put”)==0)x.put(a.substring(i+1));
       if (b.compareTo(“take”)==0) System.out.println(“Element“+x.take());
       if (b.compareTo(“top”)==0) System.out.println(“First ele“+x.top());
       if (b.compareTo(“empty”)==0) System.out.println(“System is empty”);
       if (b.compareTo(“size”)==0) System.out.println(x.size()+” ele”);
       if (b.compareTo(“print”)==0) {x.pt();System.out.println();}
     }while (b.compareTo(“exit”)!=0);
   }
}
```

1. Add one more operation “initialize”. After that the stack becomes empty.
2. Define Queue using two stacks,
3. Define priority system using two stacks. The element taken is biggest of existing elements.
4. Define queue using front and rear.
5. Define stack using two queues. An element is put in stack by putting in first queue. An element is taken from stack by taking last element of first queue. Another queue acts as auxiliary queue.
6. Define stack using single queue.
7. Using queue define priority system. In that element taken is the biggest of existing elements.
8. Define stack using priority system. Assume all elements are positive. Do not use empty().
9. Define non abstract data type “sequence”. Its operations are put(e,k), take(k), empty and print. After operations put(12,1) put(37,2) put(41,1) put(67,2) put(95,3) take(4) then sequence will look as 41,67,95,37. The element taken will be 12.

Assignment - 13 Graphics

```
import java.awt.*;
import java.io.*;
import java.lang.*;
import javax.swing.*;
import java.applet.*;
public class gr extends Applet
{  public void init()
   {  setBackground(Color.white);
      setForeground(Color.red);
   }
   public void paint(Graphics g)
   {  g.drawRect(10,100,50,70);
      g.fillOval(10,100,50,70);
      g.drawString("Kapil",50,7);
      g.drawLine(100,20,400,70);
      g.setColor(Color.blue);
      g.drawOval(100,200,50,10);
   }
}
```

In a file whose name is **gr.java** type this program. Compile it by command **javac gr.java**. In a file **a.html** type following.

```
<html>
<applet code="gr.java" height=300 width=700>
</applet>
</html>
```

Now give command **appletviewer a.html**

g.drawRect(10,100,50,70) draws a rectangle whose north west corner is (10,100) and sides are 50 and 70.

g.drawOval(10,100,50,70) draws a ellipse inside a rectangle whose north west corner is (10,100) and sides are 50 and 70.

g.drawLine(100,20,400,70) draws a line joining (100,20) and (400,70).

In the above program replace the paint method as follows:

```
          public void paint(Graphics g)
          {  String a="";
             do{try { DataInputStream o=new DataInputStream(System.in);
                      a=o.readLine();
                } catch(Exception t){}
                if (a.compareTo("hari")==0)  g.fillOval(300,100,50,70);
                if (a.compareTo("kapil")==0) g.drawRect(30,100,50,70);
                if (a.compareTo("anil")==0)  g.setColor(Color.blue);
                if (a.compareTo("ravi")==0)  break;
             }while (1==1);
             repaint();
          }
```

1. When square 12 is typed then a square of side 12 is displayed. When square 80 is typed then a square of side 80 is displayed. The north west corner is 100 100.
2. When square x is typed then a square of side 12 is displayed. The center is 100 100.
3. When ram is typed then a red circle is displayed. When gagan is typed then green circle is displayed. When bimu is typed then blue circle is displayed. The centre is (100,100). The radius is 80.
4. When rectangle a b is typed then a rectangle of length a and breadth b is displayed. The north west corner is (100,100).

5. When circle a b c is typed then a circle with center (a,b) and radius r is displayed.

```
public class gr extends Applet
{  int x=100;
   public void init()
   {  setBackground(Color.white);setForeground(Color.red);}
   public void paint(Graphics g)
   {  g.drawLine(x,20,200,70);
      try { DataInputStream o=new DataInputStream(System.in);
            String a=o.readLine();
      } catch(Exception t){}
      x=x+5; repaint();
   }
}
```

It displays a line. After pressing ENTER the line appears moving.

6. Write a program, which initially displays a line (200,100),(200,250). After pressing ENTER the line rotates by 10 degree (anticlockwise) around (200,250). After pressing ENTER 9 times the line will become (50,250),(200,250) [Hint: (int)Math.sqrt(q) and Math.sin(q)

Assignment – 14 File Handling

```
import java.io.*;
class hari
{
    public static void main(String arguments[])
    {
        String temp="",fileName="st";
        try{
          FileReader fr= new FileReader(fileName);  open a new file stream
          BufferedReader br= new BufferedReader(fr);   wrap it in a buffer
          while((temp=br.readLine())!=null)
              System.out.println("> "+temp);
          fr.close();
        }
        catch(Exception e)
        { System.out.println("Error is "+e); }
    }
}
```

The above program reads from file whose name is "st"

```
      String fileName = "abc";
      BufferedWriter out = new BufferedWriter(new FileWriter(fileName),8);
      out.write("Gopal is good");
      out.close();
```

The above program writes into a file whose name is "abc"

1. Write program, which reads 10 numbers from a file and outputs their sum.
2. Write program, which reads a file name (from key board) and in that file writes ABCD..Z.
3. Write program, which reads a string. The string has a word and a number. Let the string be “abc 7”. In file abc1 1,2,3,..,10 is written. In file abc2 2,4,6,..,20 is written. …… In file abc7 7,14,21,..,70 is written. Assume that the number of the string is less than 7.
4. Define method read(x,y). Here x is a string and y is a number. It returns yth letter (char) of the file whose name is x. Assume that file x has only one string. Let contents of a file “abc” are “hariisgood”. The method call t=read(“abc”,6) will make t=’g’. Call read(“abc”,0) will return h.
5. Define method write(x,y,z). Here x is a string, y is a number and z is a char. It modifies yth letter of the file (name x) as z. Assume that the file x has only one string. In the context of above file the system call write(“abc”,4,’m’) will modify the file contents as “harimsgood”.
6. Define methods move(x), read(), write(y). Here x is a number and y is a letter. All the methods are in the context of a file whose name is “abc”. The method move(x) will move the file pointer at xth location. The read() will return the letter pointed by the file pointer. The method write(y) will write the letter y at position pointed by the file pointer. Let the contents of file “abc” are “hariisgood” then move(5);t=read();move(7);write(t) will make the file contents as “hariisgsood”
7. Define method seek(x,y). Here x and y are numbers. y is either 0, 1 or 2. seek(x,0) is equivalent to move x. seek(x,2) moves file pointer at xth location from last. seek(x,1) advances file pointer by x letters. seek(7,2); seek(3,1); t=read() will make t=s.
8. Using method seek and read write program to print the contents of a file in opposite order.
9. Modify read and write methods. Here file pointer is advanced after reading and writing. Hence first 3 letters of a file can be read as move(0);p=read();q=read();r=read();
10. Using read, write and seek write program to print file contents.
11. Using method seek and read write program to print the contents of a file in opposite order.

12. Using read, write and seek write program to increment every letter of file. Assume that letter z is not present. Let initial file contents are “hariisgood” then final file contents will be “ibsjjthppe”.
13. Define method seek(s,x,y), read(s), write(s,z). s is string, x is number, y is 0, 1 or 2, z is char. ‘s’ is name of file. Corresponding operations are done in file ‘s’

Assignment - 15 Event Handling

a. Mouse

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class muse extends Applet implements MouseListener, MouseMotionListener
{   int p,q;
    public void init( ){   addMouseListener(this);addMouseMotionListener(this);    }
    public void mouseClicked(MouseEvent k) { p=k.getX( );q=k.getY( );}
    public void mouseEntered(MouseEvent k) {}
    public void mouseExited(MouseEvent m) { repaint( ); }
    public void mousePressed(MouseEvent n) {}
    public void mouseReleased(MouseEvent k){}
    public void mouseDragged(MouseEvent p){}
    public void mouseMoved(MouseEvent e){}
    public void paint(Graphics f){ f.drawOval(p,q,10,10); }
}
```

When mouse is moved out of applet a circle is displayed. It is at location of last mouse click

```
public class muse extends Applet implements MouseListener, MouseMotionListener
{  int x[]=new int[20],y[]=new int[20],size=0;
    public void mouseClicked(MouseEvent k)
            { x[size]=k.getX( ); y[size]=k.getY( );size++;repaint( );}
    public void paint(Graphics f) {  for (int i=0;i<size;i++) f.drawOval(x[i],y[i],10,10); }
```

} When mouse is clicked a circle is displayed. Earlier circles are also displayed.

```
    int p,q,k;
    public void mouseClicked(MouseEvent g)  {  p=g.getX( );q=g.getY( );k=1;repaint( );}
    public void mousePressed(MouseEvent n) {  p=n.getX( );q=n.getY( );k=2;repaint( ); }
    public void paint(Graphics f)
    {   int i; if (k==1) f.drawOval(p,q,10,10); if (k==2) f.drawOval(p,q,100,100);   }
```

When mouse is clicked a small circle is drawn. When it is pressed a big circle is drawn

1. When mouse is clicked a circle (radius 70) is displayed. The center is the place of click.
2. Circle when mouse pressed. Rectangle when mouse released.
3. Circle when mouse released at the place where the mouse was pressed.
4. Line when mouse is released. The end points are places of mouse press and release.
5. Rectangle when mouse is released. The end points are places of mouse press and release. Assume that location of mouse release is east south of the place of mouse press.
6. Do above problem, when no such assumption is made.
7. Circle when mouse is released. Centre is the place of mouse pressed. The radius is the distance between mouse pressed and mouse released.
8. Circle when mouse clicked. (A) The location is the place where the mouse was previously clicked. (B) The location where mouse was clicked two steps back.
9. Line is drawn from the place of mouse pressed to mouse dragged.
10. Line is drawn from the place of mouse pressed to mouse released.
11. All such lines are displayed.
12. Initially a large number of rectangles are displayed. Their north west corners are (5,10i) i=1..30. Their length and breath are 100 and 8 respectively. When mouse is clicked inside a rectangle then it disappears.
13. Initially a large number of circles (radius 20) are displayed. Their centers are (5,40i) i=1..15. When mouse is clicked inside a circle then it disappears.

14. Initially a large number of circles (radius 20) are drawn. Their centers are (0,0),(50,0),(100,0), ... (500,0),(0,70),(50,70),..(500,70),(100,140),..(500,140),..,(500,700). When mouse is clicked then nearest circle is removed.
15. Do above problem when a circle disappears when mouse is clicked inside it.

b. Keyboard

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class key extends Applet implements KeyListener
{  String msg= “”;
   public void init()
   {  addKeyListener(this);requestFocus();}
   public void keyPressed(KeyEvent ke)
   {  int key=ke.getKeyCode();
      if ((key>30)&&(key<122)) msg=msg+(char)key;
      switch(key)
      {
         case KeyEvent.VK_F1:msg=msg.substring(1);break;
         case KeyEvent.VK_F2:msg=msg.substring(0,msg.length()-2);break;
         case KeyEvent.VK_F3:repaint();break;
         case KeyEvent.VK_PAGE_DOWN:msg +=”<pgDn>”;break;
         case KeyEvent.VK_PAGE_UP:msg +=”<pgUp>”;break;
         case KeyEvent.VK_LEFT:msg +=”<Left Arrow>”;break;
         case KeyEvent.VK_RIGHT:msg +=”<right arrow>”;break;
         case KeyEvent.VK_ENTER:msg +=’,’;break;
      }
   }
   public void keyReleased(KeyEvent ke)
   {System.out.println(msg+ke.getKeyCode());}
   public void keyTyped(KeyEvent ke){}
   public void paint(Graphics g)
   { String a,b,c;int I,j;
     g.drawString(msg,50,10); a=”ttttt”+msg;i=a.length();
     if (a.charAt(i-2)==’C’) g.drawOval(50,60,10,20);
   }
}
```

1. Modify above program so that old string vanishes when ENTER is typed.
2. Modify above program so that output is updated after typing every letter.
3. When circle is typed then a circle is drawn.
4. When “circle 10” is typed then a circle of radius 10 is drawn.
5. When “circle 30 40 10” is typed then a circle of centre (30,40) and radius 10 is drawn.
6. When “circle red” is typed then a circle of red color is drawn.
7. When circle is typed then circle is drawn. When rectangle is typed then rectangle is drawn. And so on. Also specify their parameters.
8. When F_1 is pressed a circle is drawn. When F_2 is pressed a rectangle is displayed.
9. When 5 is typed a circle of 50 is displayed. When 7 is typed a circle of radius 70 is displayed.
10. A cursor is displayed. It can be modified using arrow keys.

```
   Public void init()
  { setBackground(Color.cyan);setForeground(Color.red);a=”Ram is good”;}
  public void paint(Graphics g)
  { g.drawString(a,10,30);g.drawLine(100,20,400,70);
    g.setColor(Color.blue);g.drawString(“kapil is good”,10,85);
    g.drawRect(90,100,500,300);g.fillRect(700,100,500,300);
    g.drawOval(100,500,500,300);g.fillOval(700,500,500,300);
  }
```