JAVA (Overview)

Type following program in a file named a.java. Compile the program by javac a.java. Execute the programs using java ram. It input is a string. The first two letters of the string are deleted. [Input "prasad" Output "asad"]. In string "prasad" 0th letter is 'p'. 1st letter is 'r'. 5th letter is 'd'. substring(2) retains 2nd letters onward. [substring(3,6) retains 3rd, 4th, and 5th letters only ("omprakesh" → "rak")].

```
import java.io.*;
class ram
{ public static void main(String args[]) throws Exception
  { DataInputStream t=new DataInputStream(System.in); String k,p; k=t.readLine();
    p=k.substring(2);System.out.println(p);
  }
}
```

1. Write program, which reads two strings. Deletes 0th letters of both strings and joins them. Input ramesh and hari (in different lines) output ameshari [Hint: p=q+r will join strings q and r and store it in p].
2. Reads a string and deletes 2nd and 3rd letter. [Hint: substring(0,2)+substring(4)]. Ramprasad→Rarasad
3. Exchange 2nd and 5th letters. Ramprasad→Raaprmsad

   int m;m=Integer.parseInt(t.readLine()); m=m*2; System.out.println(m); Read a number and double it.
4. Read a number (x) and a string. Output xth letter of the string. 3 ramesh→e

   The following program reads a number and doubles it. It considers the number of first and second digit only. It doubles that. A number can be converted (type conversion) to a string by appending null string. Input 474587 → 949174 → "949174" → "49" → 49 → 98. int a;String b=a+"";Converts integer into String

   int m;m=Integer.parseInt(t.readLine( ));m=m*2; String p=m+"";
   p=p.substring(1,3); m=Integer.parseInt(p); m=m*2; System.out.println(m);
5. Read a number, double it and delete its second digit. Input 2745 output 540.
6. Read a number, find its first digit (x). Now find xth digit. Input 478293 output 9.

   a=t.readLine();b=t.readLine(); int i=a.compareTo(b);System.out.println(i);

   Read two strings. Output some positive number if first string is lexicographically bigger. Output some negative number if second string is bigger. Output is 0 if both are same.
7. Read three strings and output the lexicographically biggest string.
8. Read a string and output its last letter. Hint: i=a.length(); It finds the length of the string.

   char b;a=t.readLine();b=a.charAt(2); System.out.println(b); Letter at location 2. e.g. i/p qwert o/p e

   int i=a.indexOf('x'); System.out.println(i); Location of first 'x' If no 'x' then –1 wedxtyhxu→3. abcd→-1
9. Write program to find letter immediately after 1st 'x'. [Do not use substring] Input abxedr output 'e'.
10. Location of second 'x'. i/p tuxrxyux o/p4. (A)Assume at least 2 x's. (B)When 2nd x is absent then o/p "abs".

    b="";for(i=1;i<=4;i++){j=a.indexOf('x');b=a.substring(0,j);a=a.substring(j+1);System.out.println(b+",");}

    This program prints first 4 substrings before x input pwsxtxxaxrdexgxhi output pws,t,,a,
11. Write program to find the string after 4th x [rdexgxhi]. (A) The string after last 'x' [hi]

    float x,y,k; a=t.readLine(); a=a.trim(); int 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; System.out.println(k+" "+x*y);

    Read two numbers and outputs their sum and product. Both numbers are given in same line.
12. Read 3 numbers in same line and output their sum. (A) The sum of all numbers given in a line.

    The following program reads a string. Converts it into an array of letters (using getBytes). Now 0th letter is incremented. The array of letters is again converted into string.

```
class gopal
{ public static void main(String args[]) throws Exception
  { DataInputStream t=new DataInputStream(System.in); String m;m=t.readLine();
    byte a[]=new byte[100];a=m.getBytes(); a[0]++; m=new String(a); System.out.println(m);
  }
}
```

13. Read a string and exchange its 0th and 1st letters. Do not use substring.
14. Read a string and delete its 0th letter. Do not use substring.
15. Read a string and find how many letters are capital. [ASCII code between 65 and 90].

Thread

```
class xyz implements Runnable
{  public void run( )
   { int i;
     for (i=0;i<5;i++)
     { System.out.print(i);
       try{Thread.sleep(1000);}
       catch(Exception e){}
     }
   }
}
```

```
class kapil
{  public static void main(String ar[]) throws Exception
   { xyz k;Thread a,b;
     k=new xyz( );a=new Thread(k);
     b=new Thread(k);
     a.start( );
     System.out.print("x");
   }
}
```

The possible outputs of above program are x01234, 0x1234.

When a.start( ) is replaced by a.run( ) then only one output (01234x) is possible.

[a.start() means initiate execution of 'a'. The next instruction may start execution even if 'a' is not over. a.run() means that the next instruction will start only when 'a' is over.]

a.start( );b.start( ) then possible outputs are x0011223344, 0x011223344, or 00x11223344

a.run( );b.start( ) outputs 01234x01234 or 012340x1234

a.start( );b.run( ) outputs 0011223344x or 001122334x4 (last 4 by a, second last by b)

1. Write program to output 0123x4 [a.start; sleep(3500);print(x)]
   Here print("x") is System.out.print("x"); sleep(3500) is try{Thread.sleep(3500);} catch(Exception e){}
2. Write program to output 010213243x4. [a.start,sleep,b.start,sleep]
3. Program to output 0x1x2x3xx4 a.start;for(i=1 to 6){sleep;print(x);}
4. Program to print (A)01x2x3xx4x (B)01x2x34x5x6x78x9xx (C)xx1x2xx3x4x5xx6x789
5. Write main program as following. After every 2.4 seconds a new thread is created.
   for (i=1 to 10) { print("x"); k=new xyz( ); a=new Thread(k); a.start( ); sleep(2400); }
   The output is x012x03142x03142x03142x.... x012x03142x03142x03142x...
   When for loop in xyz is (i=0 to 9) then output is x012x03142x05316427x0538164927x0
6. Define another class pqr. It prints letters A..Z [Hint: System.out.print((char)i); sleep(200)].
   Write main program: xyz k; pqr t; Thread a,b; k=new xyz( ); t=new pqr( );
   a=new Thread(k); b=new Thread(t); a.start( );sleep(700);b.run( );print("ram");
   Output: 0AB1CDEFG2HIJKL3MNOPQ4RSTUV5WXYZram6789
7. Modify above to print: 0ABCD1EF2GH3I4J5K6L7M8N9O..Z[sleep(100*(i-65)) in pqr]
8. Write a program, which creates threads in an infinite loop. After every 4999 milli seconds a new thread is created. Every thread prints sequence abcdef.... Time difference between consecutive letters is 1 second. The output looks as following:
   a,b,c,d,e,af,bg,ch,di,ej,afk,bgl,chm,din,ejo,afkp,bglq,chmr,dins,ejot,afkpu (,) comma not printed.
9. Modify above so that (,) is also printed. [A thread prints (,) at t=500, 1500, 2500, ...]
10. Modify above to print (a)(b)(c)(d)(e)(af)(bg)(ch)(di)(ej)(afk)(bgl)(chm)(din)(ejo)(afkp)(bglq)(chmr)(dins)(ejot)(afkpu)

Thread

```
#include<stdio.h>
#include<pthread.h>
void *f(void *x)
{ int i; sleep(1);
  for(i=1;i<=5;i++)
  { printf("%d\n",i*10);
    sleep(2);
  }
}
```

```
main()
{ pthread_t g;int j;
  pthread_create(&g,NULL,f,NULL);
  for(j=1;j<=3;j++)
  { printf("%d\n",j+346);sleep(2); }
  pthread_join(g,NULL);
  printf("anil\n"); sleep(8);
}
Compile: gcc a.c -lpthread
```

(A)	(B)	(C)	(D)
347	347	347	347
10	10	10	10
348	348	348	348
20	20	20	20
349	349	349	349
30	30	30	30
40	anil	anil	anil
50	40		40
anil	50		

(A)Output of above (B)without join (C)no join no sleep(8) (D)no join and sleep(8) replaced by sleep(2)

```
void *f(void *x)
{ while(1)
  { sleep(1);
    printf("%d\n",*(int*)x*3);
  }
}
```

```
main()
{ pthread_t g;int p; p=72;
  pthread_create(&g,NULL,f,&p );
  scanf("%d",&p);
  sleep(10);
}
```

output 216
repeatedly
gap of 1 sec
till input is given
3*input(10 times)