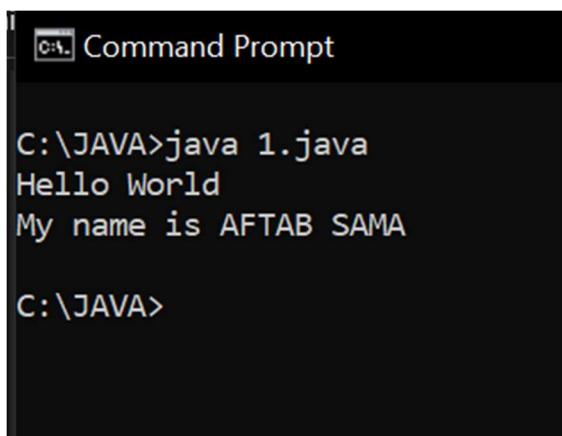


JAVA Practical

PRACTICAL 1

- 1) WAP to print Hello World and your name in a new line

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
        System.out.println("My name is AFTAB SAMA");  
    }  
}
```

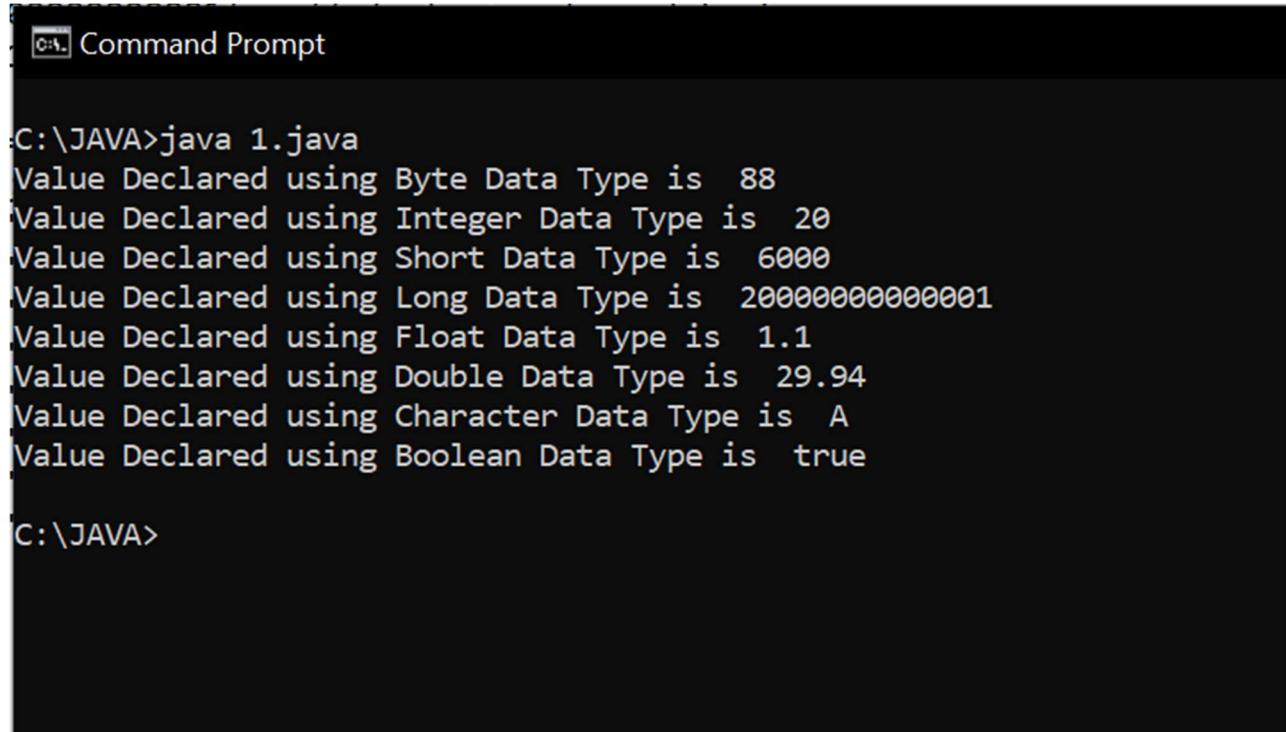


The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "C:\JAVA>java 1.java" is entered, followed by two lines of output: "Hello World" and "My name is AFTAB SAMA". The prompt "C:\JAVA>" appears again at the bottom.

```
C:\JAVA>java 1.java  
Hello World  
My name is AFTAB SAMA  
C:\JAVA>
```

- 2) WAP to use primitive datatypes in Java

```
public class main {  
    public static void main (String[] args) {  
  
        byte byteData= 88; //declaring byte data type  
        int intData= 20; // declaring integer data type  
        short shortData= 6000; //declaring short data type  
        long longData = 20000000000001l; // declaring long data type  
        float floatdata= 1.1f; // declaring float data type  
        double doubleData = 29.94d; // declaring double data type  
        boolean booleanData= true; //declaring boolean data type  
        char charData = 'A'; // declaring character data type  
        System.out.println("Value Declared using Byte Data Type is " + byteData);  
        System.out.println("Value Declared using Integer Data Type is " + intData);  
        System.out.println("Value Declared using Short Data Type is " + shortData);  
        System.out.println("Value Declared using Long Data Type is " + longData);  
        System.out.println("Value Declared using Float Data Type is " + floatdata);  
        System.out.println("Value Declared using Double Data Type is " +  
        doubleData);  
        System.out.println("Value Declared using Character Data Type is " +  
        charData);  
        System.out.println("Value Declared using Boolean Data Type is " +  
        booleanData);  
  
    }  
}
```



Command Prompt

```
C:\JAVA>java 1.java
Value Declared using Byte Data Type is 88
Value Declared using Integer Data Type is 20
Value Declared using Short Data Type is 6000
Value Declared using Long Data Type is 2000000000001
Value Declared using Float Data Type is 1.1
Value Declared using Double Data Type is 29.94
Value Declared using Character Data Type is A
Value Declared using Boolean Data Type is true

C:\JAVA>
```

3) WAP to use arithmetic operators in Java

```
class Main {
    public static void main(String[] args) {
        // declare variables
        int a = 12, b = 5;

        System.out.println(" a = 12, b = 5 ");
        // addition operator
        System.out.println("a + b = " + (a + b));

        // subtraction operator
    }
}
```

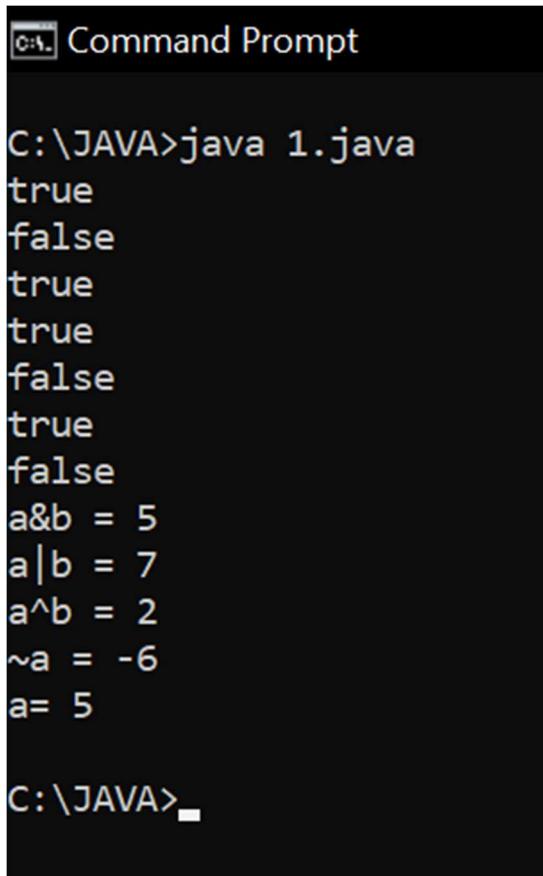
```
System.out.println("a - b = " + (a - b));  
  
// multiplication operator  
System.out.println("a * b = " + (a * b));  
  
// division operator  
System.out.println("a / b = " + (a / b));  
  
// modulo operator  
System.out.println("a % b = " + (a % b));  
}  
}
```

```
Command Prompt  
C:\JAVA>java 1.java  
a = 12, b = 5  
a + b = 17  
a - b = 7  
a * b = 60  
a / b = 2  
a % b = 2  
C:\JAVA>
```

- 4) Write a program to use logical & bitwise operators in Java

```
class Main {  
    public static void main(String[] args) {  
  
        // && operator  
        System.out.println((5 > 3) && (8 > 5)); // true  
        System.out.println((5 > 3) && (8 < 5)); // false  
  
        // || operator  
        System.out.println((5 < 3) || (8 > 5)); // true  
        System.out.println((5 > 3) || (8 < 5)); // true  
        System.out.println((5 < 3) || (8 < 5)); // false  
  
        // ! operator  
        System.out.println(!(5 == 3)); // true  
        System.out.println(!(5 > 3)); // false  
  
        int a = 5;  
        int b = 7;  
  
        // bitwise and  
        // 0101 & 0111=0101 = 5  
        System.out.println("a&b = " + (a & b));  
  
        // bitwise or  
        // 0101 | 0111=0111 = 7  
        System.out.println("a|b = " + (a | b));
```

```
// bitwise xor  
// 0101 ^ 0111=0010 = 2  
System.out.println("a^b = " + (a ^ b));  
  
// bitwise not  
// ~0101=1010  
// will give 2's complement of 1010 = -6  
System.out.println("~a = " + ~a);  
  
// can also be combined with  
// assignment operator to provide shorthand  
// assignment  
// a=a&b  
a &= b;  
System.out.println("a= " + a);  
  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "C:\JAVA>java 1.java". The output displayed is:

```
C:\JAVA>java 1.java
true
false
true
true
false
true
false
a&b = 5
a|b = 7
a^b = 2
~a = -6
a= 5
```

The prompt "C:\JAVA>" is visible at the bottom.

5) WAP to use relational operators

```
class Main {
    public static void main(String[] args) {
        int a = 7, b = 11;
        System.out.println("a is " + a + " and b is " + b);

        // == operator
        System.out.println(a == b); // false
```

```
// != operator  
System.out.println(a != b); // true  
  
// > operator  
System.out.println(a > b); // false  
  
// < operator  
System.out.println(a < b); // true  
  
// >= operator  
System.out.println(a >= b); // false  
  
// <= operator  
System.out.println(a <= b); // true  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "java 5.java" is run, followed by several println statements. The output is:

```
C:\JAVA>java 5.java  
a is 7 and b is 11  
false  
true  
false  
true  
false  
true  
C:\JAVA>
```

6) WAP to use Ternary operator & show the short circuit operators working

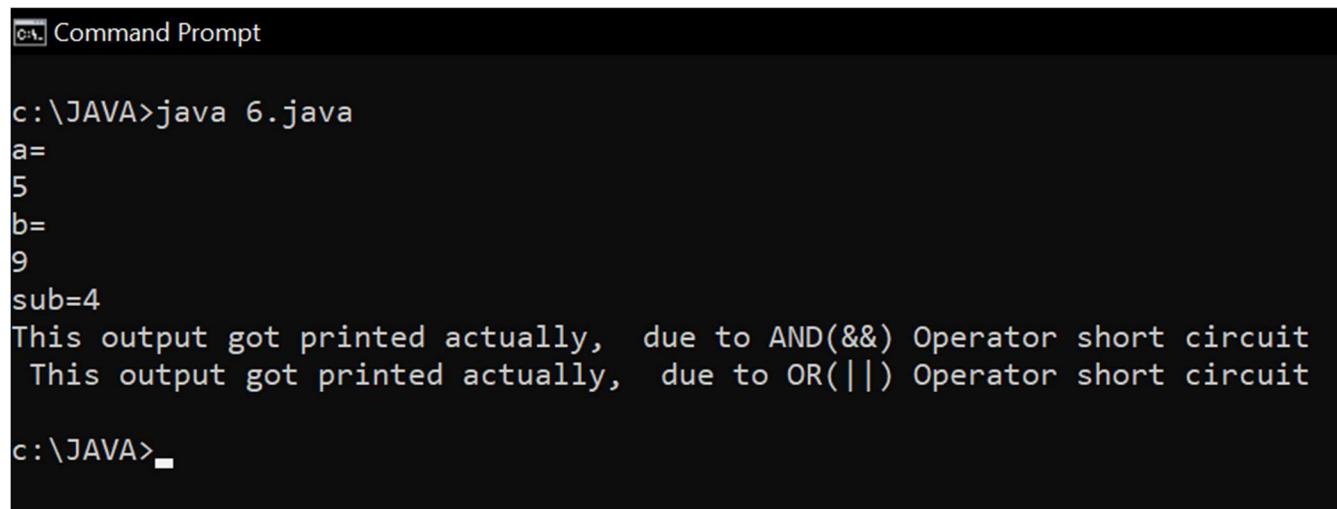
```
import java.util.Scanner;

public class Ternary {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.println("a= ");
        int a=sc.nextInt();
        System.out.println("b= ");
        int b=sc.nextInt();
        int c=(a>b) ? a-b : b-a ;
        System.out.println("sub="+c);

        if (false && true && true) {
            System.out.println("This output will not be printed");
        }
        else {

            System.out.println("This output got printed actually, due to AND(&&) Operator short circuit");
        }
        if (true || false || false) {
            System.out.println(" This output got printed actually, due to OR(||) Operator short circuit");
        }
        else {
```

```
        System.out.println("This output will not be printed");  
    }  
  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "java 6.java". The output displayed is:
a=
5
b=
9
sub=4
This output got printed actually, due to AND(&&) Operator short circuit
This output got printed actually, due to OR(||) Operator short circuit
c:\JAVA>

PRACTICAL 2

1) Write a program to display a following pattern.

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

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
  
        int i,j,k,n;  
        Scanner sc=new Scanner(System.in);  
        System.out.println("n=");  
        n=sc.nextInt();  
  
        for(i=0;i<=n;i++){  
  
            for(j=0;j<n-i;j++){  
                System.out.print(" ");  
            }  
            for(k=i;k>0;k--){  
                System.out.print("* ");  
            }  
        }  
    }  
}
```

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

```
Command Prompt  
c:\JAVA>java pyramid.java  
n=  
3  
*  
* *  
* * *  
c:\JAVA>
```

2) Convert decimal number to binary number

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        int i,j,d,k,b;  
        Scanner sc=new Scanner(System.in);  
  
        System.out.println(" Enter decimal number =");
```

```
d=sc.nextInt();

b=0;

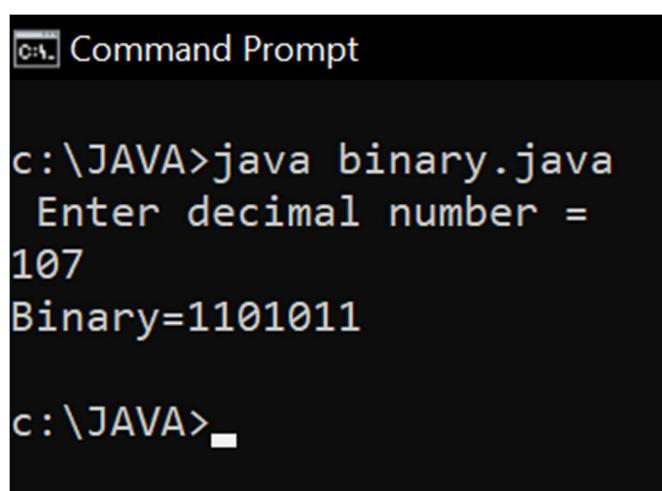
for(i=1;d>=1;i=i*10){

    j=d%2;
    d=d/2;
    b=b+(i*j);

}

System.out.println("Binary="+b);

}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "c:\JAVA>java binary.java" is entered, followed by "Enter decimal number =". The user inputs "107", and the program outputs "Binary=1101011". The prompt then changes to "c:\JAVA>".

```
c:\JAVA>java binary.java
Enter decimal number =
107
Binary=1101011
c:\JAVA>
```

3)To compute the sum of digits of an integer

```
import java.io.*;
import java.util.Scanner;

class digitsum {

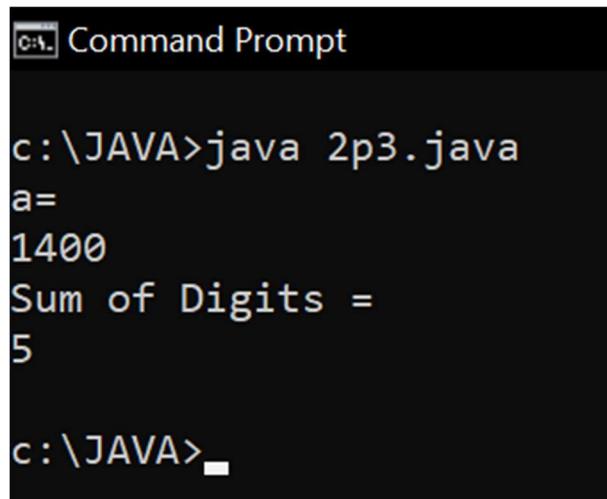
    static int getSum(int n)
    {
        int sum = 0;

        while (n != 0) {
            sum = sum + n % 10;
            n = n / 10;
        }

        System.out.println("Sum of Digits = ");
        return sum;
    }

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

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



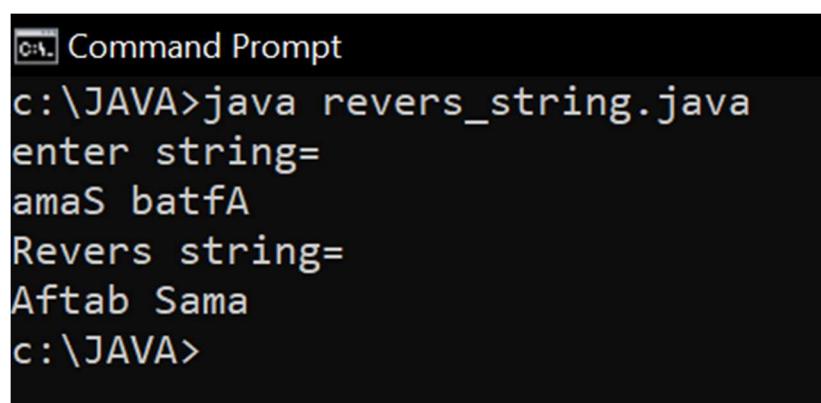
The screenshot shows a Windows Command Prompt window titled "Command Prompt". The path "c:\JAVA>" is visible at the top. The user has run the command "java 2p3.java". The application's output is displayed below, showing the input "a=" followed by "1400", then "Sum of Digits =", and finally "5". The prompt "c:\JAVA>" is shown again at the bottom.

```
c:\JAVA>java 2p3.java  
a=  
1400  
Sum of Digits =  
5  
c:\JAVA>
```

4)WAP to reverse a string

```
import java.util.Scanner;  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner sc=new Scanner(System.in);  
        String s1;  
        int i=0,n;  
        System.out.println("enter string= ");  
        s1=sc.nextLine();  
        n=s1.length();  
        char[]a= new char[n];  
        for(i=0;i<n;i++)
```

```
{  
    a[i]=s1.charAt(i);  
  
}  
System.out.println("Revers string= ");  
for(int j=n-1;j>=0;j--)  
{  
    System.out.print(a[j]);  
}  
  
}  
}  
}
```



The screenshot shows a Command Prompt window with the following text:

```
c:\ Command Prompt  
c:\JAVA>java revers_string.java  
enter string=  
amaS batfA  
Revers string=  
Aftab Sama  
c:\JAVA>
```

5)WAP to count letters spaces number and other special characters

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        String s1;
        int i,s=0,c=0,sch=0,n=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("enter string=");
        s1=sc.nextLine();
        for(i=0;i<s1.length();i++)
        {
            if(s1.charAt(i)==' ') { s++; } else
            if(s1.charAt(i)>='a'&&s1.charAt(i)<='z'||s1.charAt(i)>=
            'A'&&s1.charAt(i)<='Z') { c++; } else if(s1.charAt(i)>=
            '0'&&s1.charAt(i)<='9') { n++; } else { sch++; }
        }
        System.out.println("number of space in string is "+s);
        System.out.println("number of character in string is "+c);
        System.out.println("number of special character in string is "+sch);
        System.out.println("number of digits in string is "+n);
    }
}
```

```
c:\ Command Prompt
c:\JAVA>java number_of_char.java
enter string=
Aftab Sama @#- RSU1916041 1 a
number of space in string is 5
number of character in string is 13
number of special character in string is 3
number of digits in string is 8

c:\JAVA>
```

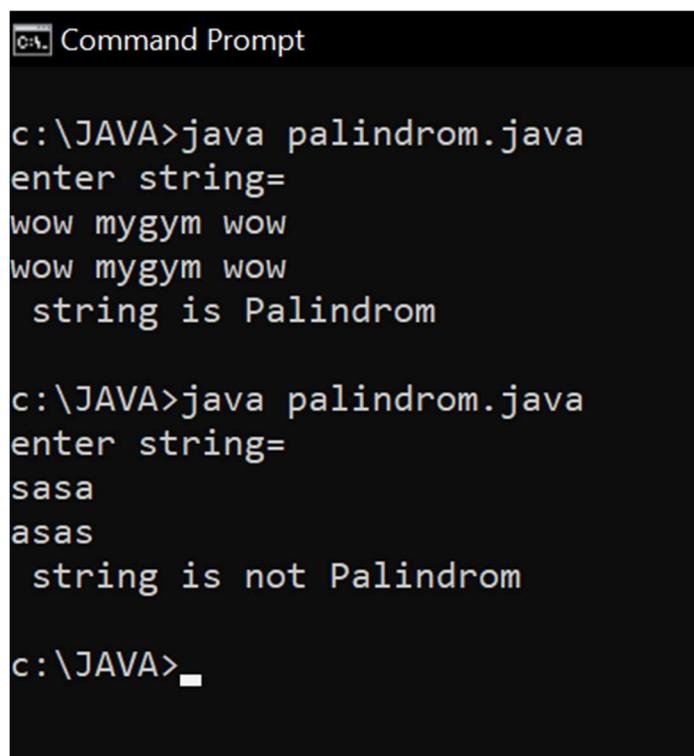
6)WAP to check palindrome of a string

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);
        String s1;
        int i=0,n;
        System.out.println("enter string= ");
        s1=sc.nextLine();
        n=s1.length();
        char[]a= new char[n];
        for(i=0;i<n;i++)
        {
            a[i]=s1.charAt(i);

        }
    }
}
```

```
for(int j=n-1;j>=0;j--)  
{  
    System.out.print(a[j]);  
}  
  
int k=0;  
  
for(int c=0;c<n;c++){  
    if(a[n-1-c]!=s1.charAt(c)){ k++; };  
}  
  
System.out.println(" ");  
  
if(k==0){ System.out.println(" string is Palindrom"); } else{  
System.out.println(" string is not Palindrom"); }  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". It displays two executions of a Java program named "palindrom.java".

In the first execution, the user enters the string "wow mygym wow" and the program outputs "string is Palindrom".

In the second execution, the user enters the string "sasa" and the program outputs "string is not Palindrom".

```
c:\JAVA>java palindrom.java  
enter string=  
wow mygym wow  
wow mygym wow  
string is Palindrom  
  
c:\JAVA>java palindrom.java  
enter string=  
sasa  
asas  
string is not Palindrom  
  
c:\JAVA>
```

PRACTICAL 3

1) Program to create and display unique 3 digits number using 1,2,3,4.
Also count how many 3 digits no.s are there

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int d[]={},i,k,k1=0,i1=0,j1=0,l=0,l1=0,j;
        System.out.println("enter d1=");
        d[0]=sc.nextInt();
        System.out.println("enter d2=");
        d[1]=sc.nextInt();
        System.out.println("enter d3=");
        d[2]=sc.nextInt();
        System.out.println("enter d4=");
        d[3]=sc.nextInt();
        for(i=0;i<2;i++){
            for(j=0;j<2;j++){
                j1=k1+1;
                for(k=0;k<6; ){
```

```
if(i1>3){ i1=0;}
```

```
if(j1>3){ j1=0;}
```

```
{
```

```
System.out.print( d[k1]);
```

```
}
```

```
{
```

```
System.out.print( d[j1] );
```

```
}
```

```
if(i1==k1){ i1++;}
```

```
if(i1==j1){ i1++;}
```

```
if(i1>3){ i1=0;} {
```

```
System.out.print( d[i1] );
```

```
}
```

```
if(k%2!=0){ j1++; }
```

```
i1++;
```

```
if(k==5){ k1++; l++; }

System.out.println(" ");
k++;
}

}

}

System.out.println("total unique number = 24");

}

}
```

```
C:\ Command Prompt
c:\JAVA>java unique3digit4.java
enter d1=
1
enter d2=
2
enter d3=
3
enter d4=
4
123
124
132
134
142
143
234
231
243
241
213
214
341
342
314
312
324
321
412
413
421
423
431
432
total unique number = 24
c:\JAVA>
```

2)WAP to print ASCII value of given char and check using function whether it is no., alphabet, space or special chars

```
import java.util.*;  
  
public class Main {  
    public static void main(String[] args) {  
        String ch; int i,j,c=0,s=0,n=0,schar=0;  
        Scanner sc=new Scanner(System.in);  
        System.out.println("enter char=");  
        ch=sc.nextLine();  
        for(i=0;i<ch.length();i++){ j=ch.charAt(i);  
            System.out.println("ASCII value of "+ch.charAt(i)+" is "+j);}  
        for(i=0;i<ch.length();i++){  
            if(Character.isDigit(ch.charAt(i))){ n++; }  
            if(Character.isAlphabetic(ch.charAt(i))){ c++; }  
            if(Character.isSpace(ch.charAt(i))){ s++; }  
            schar=ch.length()-n-c-s;  
        }  
  
        System.out.println("total number is "+n);  
        System.out.println("total space is "+s);  
        System.out.println("total special char is "+schar);  
        System.out.println("total character is "+c);  
    }  
}
```

```
c:\JAVA>java ascii_character.java
Note: ascii_character.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
enter char=
A@1- a
ASCII value of 'A' is 65
ASCII value of '@' is 64
ASCII value of '1' is 49
ASCII value of '-' is 45
ASCII value of ' ' is 32
ASCII value of 'a' is 97
total number is 1
total space is 1
total special char is 2
total character is 2

c:\JAVA>
```

3)WAP to multi-dimensional array with second dimension of sizes 3,5,2,4 and iterate to point its value

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int [][]a=new int [4][];
        int []n=new int [4]; int i,j,k;
        System.out.println("columns for row 1=");
    }
}
```

```
n[0]=sc.nextInt();
System.out.println("columns for row 2=");
n[1]=sc.nextInt();
System.out.println("columns for row 3=");
n[2]=sc.nextInt();
System.out.println("columns for row 4=");
n[3]=sc.nextInt();
a[0]= new int[n[0]];
a[1]= new int[n[1]];
a[2]= new int[n[2]];
a[3]= new int[n[3]];

System.out.println("enter value for 1 row=");
for(i=0;i<n[0];i++){
    a[0][i]=sc.nextInt();
}
System.out.println("enter value for 2 row=");
for(i=0;i<n[1];i++){
    a[1][i]=sc.nextInt();
}
System.out.println("enter value for 3 row=");
for(i=0;i<n[2];i++){
    a[2][i]=sc.nextInt();
}
System.out.println("enter value for 4 row=");
for(i=0;i<n[3];i++){
    a[3][i]=sc.nextInt();}
```

```
        }  
        System.out.println(" ");  
        for(j=0;j<4;j++){  
            for(k=0;k<n[j];k++){  
                System.out.print(" "+a[j][k]);  
            }  
            System.out.println(" ");  
        }  
    }  
}
```

```
c:\JAVA>java variablesize_array.java  
columns for row 1=  
3  
columns for row 2=  
5  
columns for row 3=  
2  
columns for row 4=  
4  
enter value for 1 row=  
1 2 3  
enter value for 2 row=  
1 2 3 4 5  
enter value for 3 row=  
1 2  
enter value for 4 row=  
1 2 3 4  
  
1 2 3  
1 2 3 4 5  
1 2  
1 2 3 4  
c:\JAVA>
```

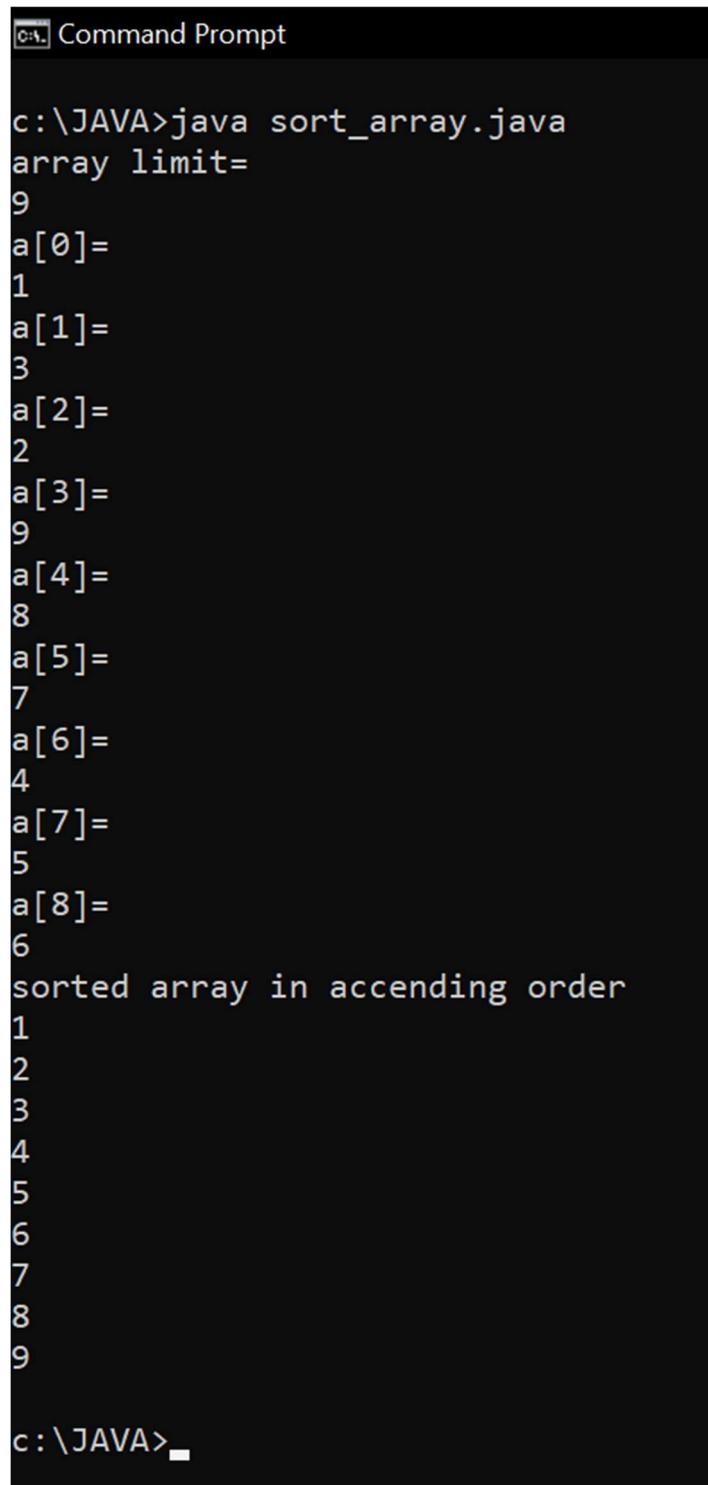
4)WAP to sort an array using minimum value comparison

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int i,j,n,x;
        Scanner sc=new Scanner(System.in);
        System.out.println("array limit=");
        n=sc.nextInt();
        int a[]={};
        for(i=0;i<n;i++){
            System.out.println("a["+i+"]=");
            a[i]=sc.nextInt();
        }
        System.out.println("sorted array in accending order");

        for(i=0;i<n;i++){
            x=a[i];
            for(j=i;j<n;j++){
                if(a[j]<x){ x=a[j]; a[j]=a[i]; a[i]=x; }

            }
            for(i=0;i<n;i++){
                System.out.println(a[i]);
            }
        }
    }
}
```

```
    }  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "c:\JAVA>java sort_array.java". The output displays the array elements from index 0 to 8, followed by a sorted array in ascending order from 1 to 9.

```
c:\JAVA>java sort_array.java  
array limit=  
9  
a[0]=  
1  
a[1]=  
3  
a[2]=  
2  
a[3]=  
9  
a[4]=  
8  
a[5]=  
7  
a[6]=  
4  
a[7]=  
5  
a[8]=  
6  
sorted array in accending order  
1  
2  
3  
4  
5  
6  
7  
8  
9  
c:\JAVA>
```

5)WAP to implement StringBuffer to increase its capacity and modify it until user insists.

```
import java.util.*;
import java.io.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int i,j,k,n,p;
        String x;
        String v;
        StringBuffer s =new StringBuffer(5);

        for(i=0; i<3; )
        {
            System.out.println(" enter 1-add string ,2-replace , 3-
delete 4-insert 0-exit ");
            n=sc.nextInt();
            if(n==1){
                System.out.println("enter string=");
                x=sc.nextLine();
                x=sc.nextLine();
                s.append(x);
            }
        }
    }
}
```

```
System.out.println(" StringBuffer="+s);
System.out.println(" StringBuffer
capacity="+s.capacity());
System.out.println(" StringBuffer size="+s.length());

i=1;
}
if(n==4){
    System.out.println("enter position");
    p=sc.nextInt();
    System.out.println("enter value");
    v=sc.nextLine();
    v=sc.nextLine();
    s.insert(p,v);

System.out.println(" StringBuffer="+s);
System.out.println(" StringBuffer
capacity="+s.capacity());
System.out.println(" StringBuffer size="+s.length());
i=1;

}
if(n==3){
    System.out.println("enter start position");
    j=sc.nextInt();
    System.out.println("enter end position");
```

```
k=sc.nextInt();
s.delete(j,k);
System.out.println(" StringBuffer="+s);
System.out.println(" StringBuffer
capacity="+s.capacity());
System.out.println(" StringBuffer size="+s.length());
i=1;
}
if(n==2)
{
System.out.println("enter start position");
j=sc.nextInt();
System.out.println("enter end position");
k=sc.nextInt();
System.out.println("enter value");
v=sc.nextLine();
v=sc.nextLine();
s.replace(j,k,v);
System.out.println(" StringBuffer="+s);
System.out.println(" StringBuffer
capacity="+s.capacity());
System.out.println(" StringBuffer size="+s.length());
i=1;
}

if(n==0){ i=4;}
```

```
}
```

```
}
```

```
c:\JAVA>java StringBuffer.java
  enter 1-add string ,2-replace , 3- delete 4-insert 0-exit
1
enter string=
Aftab
  StingBuffer=Aftab
  StingBuffer capacity=5
  StingBuffer size=5
  enter 1-add string ,2-replace , 3- delete 4-insert 0-exit
4
enter position
5
enter value
  Sama
  StingBuffer=Aftab Sama
  StingBuffer capacity=12
  StingBuffer size=10
  enter 1-add string ,2-replace , 3- delete 4-insert 0-exit
2
enter start position
5
enter end position
6
enter value
@
  StingBuffer=Aftab@Sama
  StingBuffer capacity=12
  StingBuffer size=10
  enter 1-add string ,2-replace , 3- delete 4-insert 0-exit
0

c:\JAVA>
```

6)WAP to search a value in m*n matrix

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int i,m,n,j,k,x=0,r,v;
        Scanner sc=new Scanner(System.in);
        System.out.println("row=");
        m=sc.nextInt();
        System.out.println("column=");
        n=sc.nextInt();
        int a[][]=new int [m][n];
        for(i=0;i<m;i++){
            for(j=0;j<n;j++){
                System.out.print("a["+i+"]["+j+"]=");
                a[i][j]=sc.nextInt();
            }
        }
        for(i=0;i<m;i++){
            for(j=0;j<n;j++){
                System.out.print(" " +a[i][j]);
            }
        }
    }
}
```

```
        System.out.println(" ");
    }
    for(r=0;r<3;){
        System.out.println(" ");
        System.out.println("enter 1-search 0-exit=");
        v=sc.nextInt();
        if(v==1){   System.out.println("search=");
            k=sc.nextInt();
            for(i=0;i<m;i++){
                for(j=0;j<n;j++){
                    if(a[i][j]==k){   System.out.print("match found " +
                        "a["+i+"]"+"["+j+"]="++ a[i][j] );
                        x=1;           }
                }
            }
            if(x!=1){ System.out.println("no match found"); }
        }else if(v==0){ r=5;}else{
            System.out.println("invalid input");
        }
    }
}
```

```
c:\ Command Prompt  
c:\JAVA>java search_array.java  
row=3  
column=3  
a[0][0]=1  
a[0][1]=2  
a[0][2]=3  
a[1][0]=4  
a[1][1]=5  
a[1][2]=6  
a[2][0]=7  
a[2][1]=8  
a[2][2]=9  
 1   2   3  
 4   5   6  
 7   8   9  
  
enter 1-search 0-exit=1  
search=6  
match found a[1][2]=6  
enter 1-search 0-exit=0  
  
c:\JAVA>
```

PRACTICAL 4

1)WAP to implement constructor, method overloading for class student for at least 5 attributes and 3-5 methods

```
public class student
{
    public static void student(String studentName,char grade)
    {
        System.out.println("Student name is "+studentName);
        System.out.println("Student grade is "+grade);
    }

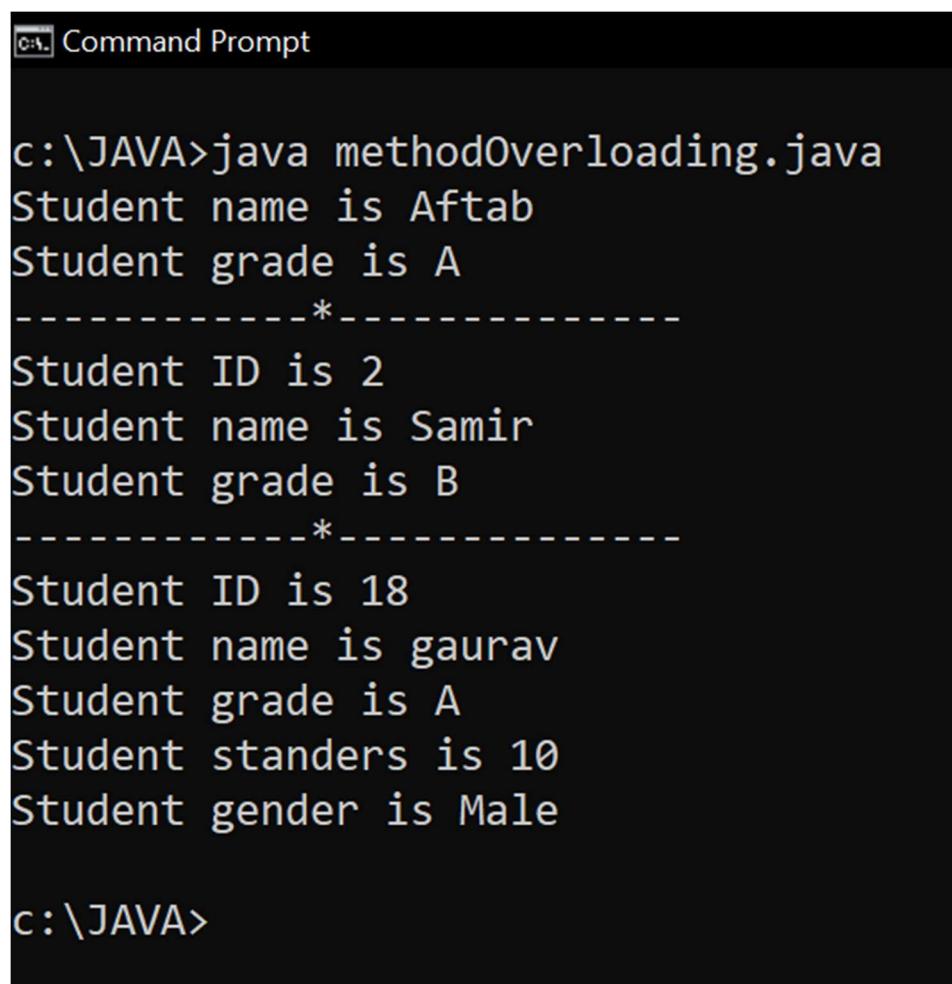
    public static void student(int id,String studentName,char grade)
    {
        System.out.println("Student ID is "+id);
        System.out.println("Student name is "+studentName);
        System.out.println("Student grade is "+grade);
    }

    public static void student(int id,String studentName,char grade,int standerd,String gender)
    {
        System.out.println("Student ID is "+id);
        System.out.println("Student name is "+studentName);
        System.out.println("Student grade is "+grade);
        System.out.println("Student standers is "+standerd);
        System.out.println("Student gender is "+gender);
    }
}
```

```
public static void main(String[]args)
{
    //student Obj=new student();
    student("Aftab",'A');
    System.out.println("-----*-----");
    student(2, "Samir", 'B');
    System.out.println("-----*-----");
    student(18, "gaurav", 'A',10,"Male");

}

}
```

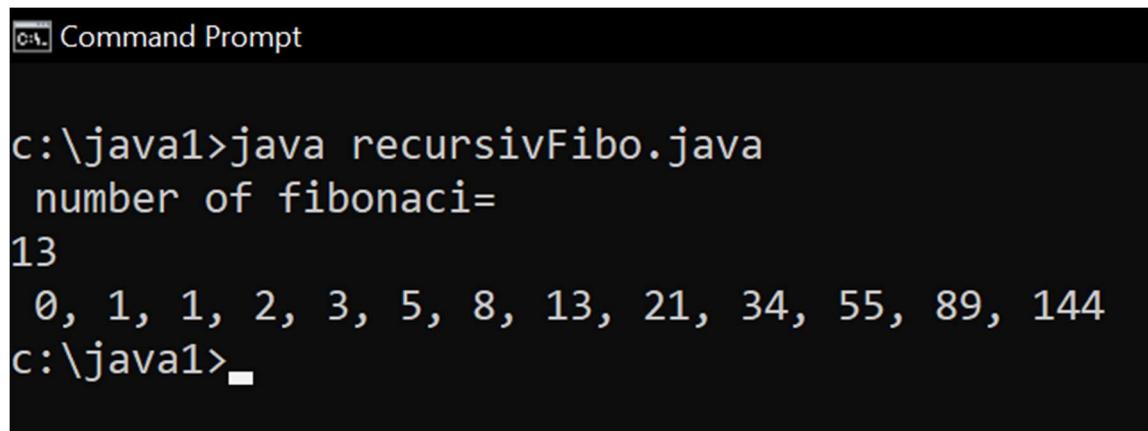


c:\JAVA>java methodOverloading.java
Student name is Aftab
Student grade is A
-----*-----
Student ID is 2
Student name is Samir
Student grade is B
-----*-----
Student ID is 18
Student name is gaurav
Student grade is A
Student standers is 10
Student gender is Male

c:\JAVA>

2)show recursion in java for Fibonacci sequence

```
import java.util.*;
public class recursivFibo{
    public static int a=1,s=0;
    public static void fibo(int n){
        if(s==0) {
            System.out.print(" 0, 1");
        }
        System.out.print(", "+(a+s));
        if(n>3){
            a=s+a; s=a-s;
            n--;
            fibo(n);
        }
    }
    public static void main(String args[]) {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println(" number of fibonaci=");
        n=sc.nextInt();
        fibo(n);
    }
}
```



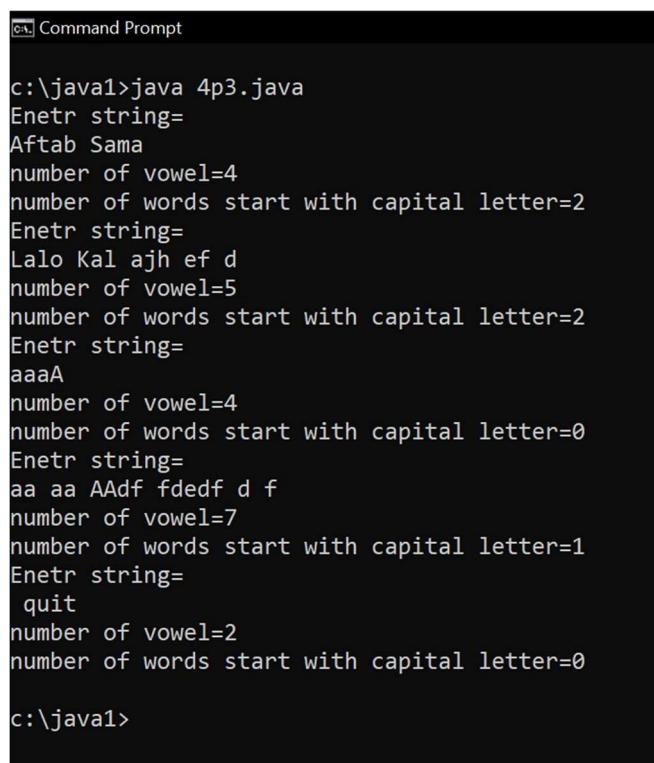
```
c:\java1>java recursivFibo.java
number of fibonaci=
13
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144
c:\java1>
```

3)create a class to take a statement as i/p and count (i)vowels of each type in it (II) words that start with capital letter. Continue till user types “quit”

```
import java.util.*;
class p3 {
    public static void main(String[] args) {
        String s;
        Scanner sc=new Scanner(System.in);
        int i=0,x,n=0,a=0,j,r;
        /*  count vowels and words that start with capital letters*/
        for (x=0;x<3 ; ) {
            System.out.println("Enetr string=");
            s=sc.nextLine();
            r=s.length();
            if ( s.length()>4&&s.charAt(r-4)=='q'&&s.charAt(r-3)=='u'&&s.charAt(r-2)=='i'&&s.charAt(r-1)=='t')
```

```
{ x=7;  
}  
  
for (i=0;i<s.length() ;i++ ) {  
    if (s.charAt(i)=='a') { a++;  
  
    }if (s.charAt(i)=='e') { a++;  
  
    }if (s.charAt(i)=='i') { a++;  
  
    }if (s.charAt(i)=='o') { a++;  
  
    }if (s.charAt(i)=='u') { a++;  
  
    }if (s.charAt(i)=='A') { a++;  
  
    }if (s.charAt(i)=='E') { a++;  
  
    }if (s.charAt(i)=='I') { a++;  
  
    }if (s.charAt(i)=='O') { a++;  
  
    }if (s.charAt(i)=='U') { a++;  
  
}
```

```
 }if (s.charAt(0)>='A'&&s.charAt(0)<='Z') { n++;  
  
 }for (j=0;j<s.length()-1 ;j++ ) {  
 if (s.charAt(j)==' '&& s.charAt(j+1)>='A'&&s.charAt(j+1)<='Z')  
 { n++;  
  
 }  
  
 }  
  
 System.out.println("number of vowel="+a);  
 System.out.println("number of words start with capital letter="+n);  
 a=0; n=0;  
  
 }  
 }  
 }
```

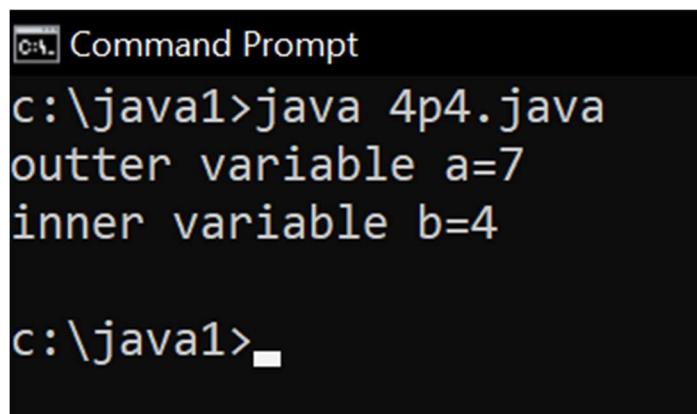


```
c:\java1>java 4p3.java  
Enetr string=  
Aftab Sama  
number of vowel=4  
number of words start with capital letter=2  
Enetr string=  
Lalo Kal ajh ef d  
number of vowel=5  
number of words start with capital letter=2  
Enetr string=  
aaaA  
number of vowel=4  
number of words start with capital letter=0  
Enetr string=  
aa aa AAdf fdedf d f  
number of vowel=7  
number of words start with capital letter=1  
Enetr string=  
 quit  
number of vowel=2  
number of words start with capital letter=0  
c:\java1>
```

4)create an outer class employee and inner class (non-static). Call the method of inner class to access details within both.

```
class main{  
    public static void main(String[] args) {  
        /*inner class to access variable with both*/  
        // Scanner sc=new Scanner(System.in);  
        outter out=new outter();  
        outter.inner inr=out.new inner();  
  
        System.out.println("outter variable a="+inr.o());  
  
        System.out.println("inner variable b="+inr.d());  
  
    }  
}  
  
public class outter{  
    int a=7;  
  
    public class inner{  
        int b=4;  
        public int d(){  
            return b;  
        }  
        public int o() {  
    
```

```
    return a;  
}  
}  
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "c:\java1>java 4p4.java" is entered, followed by the output: "outer variable a=7" and "inner variable b=4". The prompt "c:\java1>" is visible at the bottom.

```
c:\ Command Prompt  
c:\java1>java 4p4.java  
outer variable a=7  
inner variable b=4  
c:\java1>
```

5)WAP to implement Base Class (abstract) and child class Cat, Dog, Lion, Tiger to implement methods declared in base class. Also use constructors in base and derived classes

```
class p5{  
    public static void main(String[] args) {  
        base b=new cat();  
        base b1=new dog();  
        base b2=new lion();  
        base b3=new tiger();  
  
        b.draw();  
        b1.draw();
```

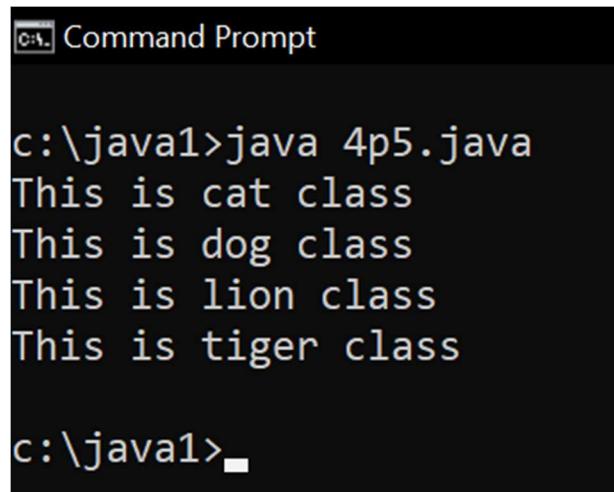
```
b2.draw();  
b3.draw();  
  
}  
}  
  
public abstract class base{  
    base(){  
  
    }  
    public abstract void draw();  
}  
  
public class cat extends base{  
    cat(){  
  
    }  
    public void draw(){  
        System.out.println("This is cat class");  
  
    }  
}  
  
public class dog extends base{  
    dog(){  
  
    }  
    public void draw(){  
        System.out.println("This is dog class");  
    }  
}
```

```
}
```

```
public class lion extends base{
    lion(){}
    public void draw(){
        System.out.println("This is lion class");
    }
}
```

```
public class tiger extends base{
    tiger(){}
    public void draw(){
        System.out.println("This is tiger class");
    }
}
```

```
}
```



```
c:\java1>java 4p5.java
This is cat class
This is dog class
This is lion class
This is tiger class

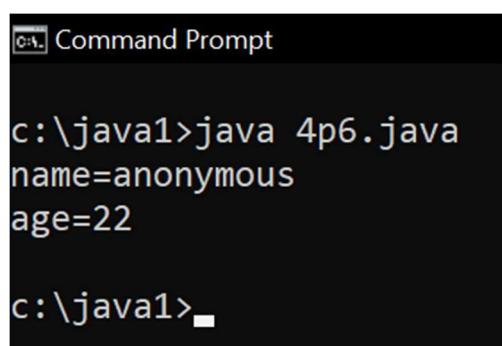
c:\java1>
```

6)WAP to implement Anonymous Inner class of employee and print its details.

```
// anonymous inner class
class p6
{
    public static void main(String[] args)
    {
        an obj=new an()
        {
            public void getan()
            {
                System.out.println("name="+s);
                System.out.println("age="+n);
            }
        };
        obj.getan();
    }
}

public abstract class an
{
    int n=22;
```

```
String s="anonymous";  
public abstract void getan();  
  
}
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:
c:\java1>java 4p6.java
name=anonymous
age=22
c:\java1>

PRACTICAL 5

1. WAP to implement Multilevel and hierarchical inheritance.

```
// multilevel inheritance and heirarchical inheritance  
import java.util.*;  
class main{  
    public static void main(String[] args) {  
        String n1,n2;  
        Scanner sc=new Scanner(System.in);  
        son s=new son();  
        daughter d=new daughter();
```

```
System.out.println("This is son : Enter name = ");
n1=sc.nextLine();

System.out.println("This is Daughter : Enter name = ");
n2=sc.nextLine();

son s1=new son(n1);
daughter d1=new daughter(n2);

}

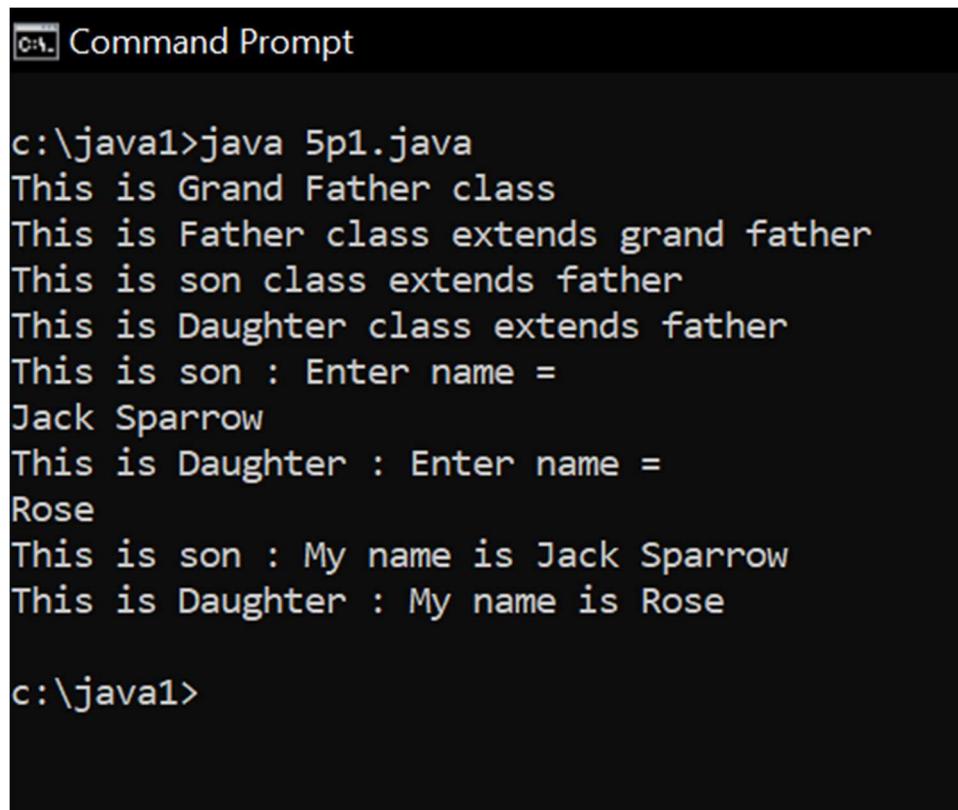
}

public class gfather{
    public void gf(){
        System.out.println("This is Grand Father class");
    }
}

public class father extends gfather{
    public void f(){
        System.out.println("This is Father class extends grand father");
    }
}

public class son extends father{
    son(){
        gf();
        f();
    }
}
```

```
        System.out.println("This is son class extends father");  
    }  
    son(String name){  
  
        System.out.println("This is son : My name is "+name);  
  
    }  
}  
  
public class daughter extends father{  
  
    daughter(){  
        System.out.println("This is Daughter class extends father");  
    }  
    daughter(String name){  
  
        System.out.println("This is Daughter : My name is "+name);  
  
    }  
}
```



```
c:\java1>java 5p1.java
This is Grand Father class
This is Father class extends grand father
This is son class extends father
This is Daughter class extends father
This is son : Enter name =
Jack Sparrow
This is Daughter : Enter name =
Rose
This is son : My name is Jack Sparrow
This is Daughter : My name is Rose

c:\java1>
```

2. Write a program that illustrates interface inheritance. Interface P12 inherits from both P1 and P2. Each interface declares one constant and one method. The class Q implements P12. Instantiate Q and invoke each of its methods. Each method displays one of the constants.

```
public class q implements p12{
    public void d1(){
        System.out.println("p1:interface p1 contains i="+i);
    }
    public void d2(){
        System.out.println("p2:interface p2 contains j="+j);
    }
}
```

```
}

public static void main(String[] args) {
    q p=new q();

    p.d1();
    p.d2();
}

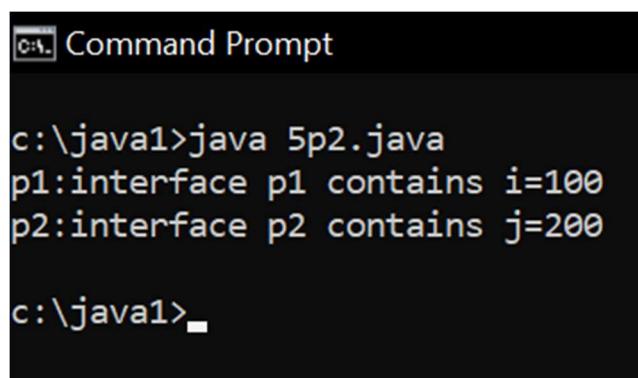
public interface p1{
    int i=100;
    void d1();
}

public interface p2{
    int j=200;
    void d2();
}

public interface p12 extends p1,p2{



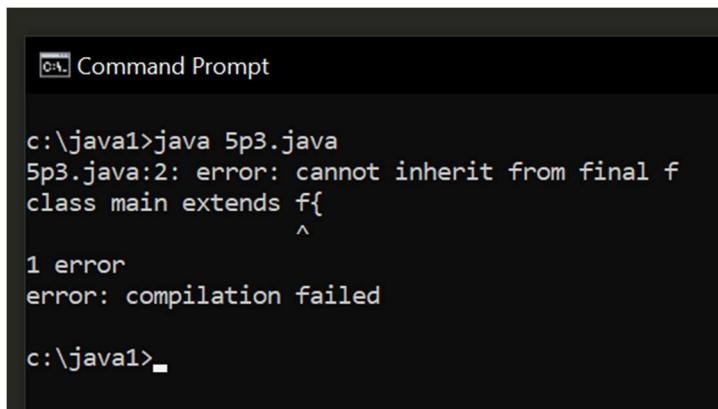
}
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "c:\java1>java 5p2.java". The output displayed is:
p1:interface p1 contains i=100
p2:interface p2 contains j=200
The prompt then returns to "c:\java1>".

3. Write a program in Java to demonstrate use of final class.

```
final class f{}  
class main extends f{  
    public static void main(String[] args) {  
  
    }  
}
```



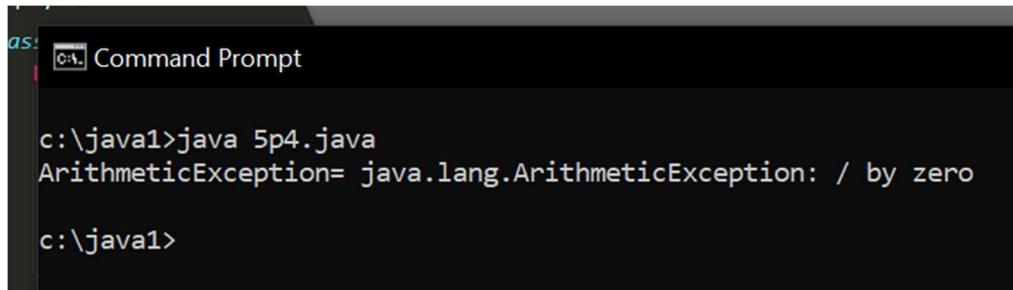
The screenshot shows a Command Prompt window with the following text:

```
c:\java1>java 5p3.java  
5p3.java:2: error: cannot inherit from final f  
class main extends f{  
      ^  
1 error  
error: compilation failed  
c:\java1>
```

4. WAP to demonstrate DivideByZero Exception in Java.

```
class p4{  
    public static void main(String[] args) {  
        int a,b,c;  
        a=55;  
        b=0;  
        try { c=a/b; }
```

```
        catch (ArithmaticException e){  
            System.out.println("ArithmaticException= "+e);  
        }  
    }  
}
```

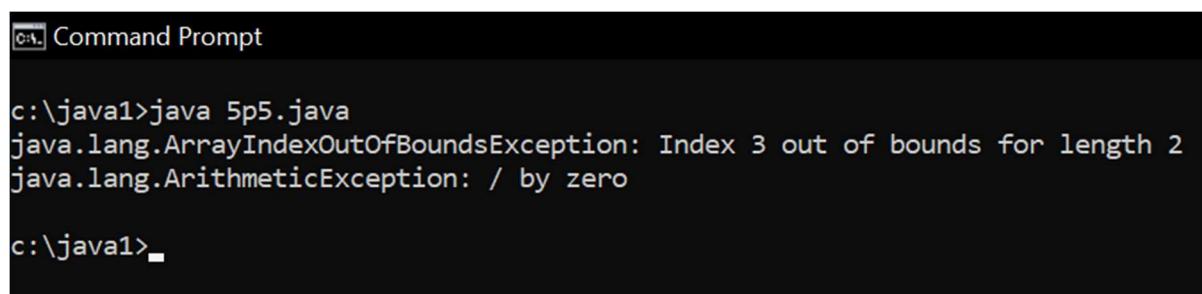


as: Command Prompt
c:\java1>java 5p4.java
ArithmaticException= java.lang.ArithmaticException: / by zero
c:\java1>

5. WAP to show nesting of try blocks using ArithmaticException and ArrayIndexOutOfBoundsException.

```
class p5{  
    public static void main(String[] args) {  
        int []a=new int[2];  
        a[0]=1;  
        a[1]=2;  
        try{  
            try{  
                a[3]=4;  
            }catch(ArrayIndexOutOfBoundsException e2){  
                System.out.println(e2);  
            }  
            a[2]=3/0;  
            a[1]=1/0;  
        }  
    }  
}
```

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



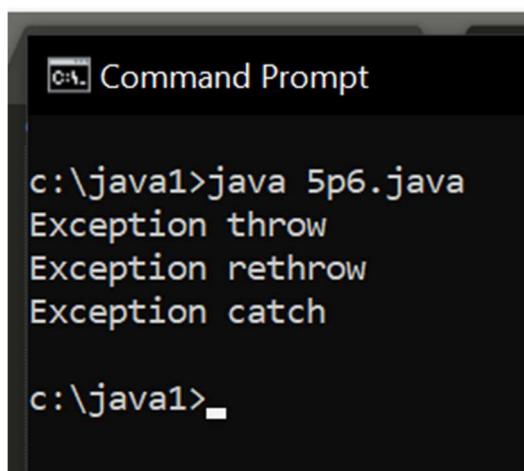
The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "c:\java1>java 5p5.java". The output displays two exceptions: "java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2" and "java.lang.ArithmetricException: / by zero". The prompt "c:\java1>" is visible at the bottom.

```
c:\java1>java 5p5.java  
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2  
java.lang.ArithmetricException: / by zero  
c:\java1>
```

6. WAP to show execution of throw and rethrow.

```
class p6{  
    public static void t() throws Exception{  
        try{  
            System.out.println("Exception throw");  
            throw new Exception("Exception throw");  
        }
```

```
        }catch(Exception e){  
            System.out.println("Exception rethrow");  
            throw e;  
        }  
    }  
    public static void main(String[] args) {  
  
        try{  
            t();  
        }catch(Exception e){  
            System.out.println("Exception catch ");  
        }  
    }  
}
```

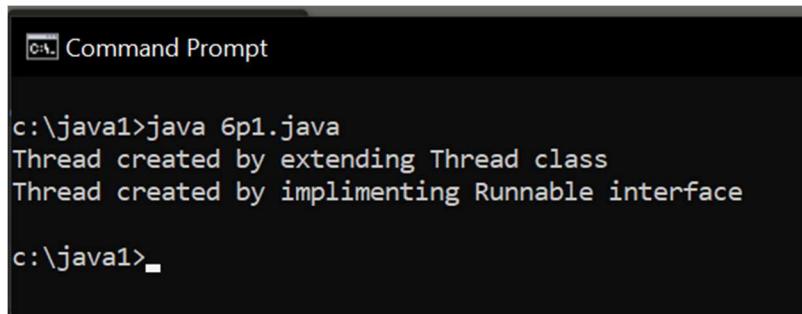


The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is "c:\java1>java 5p6.java". The output displayed is:
Exception throw
Exception rethrow
Exception catch
The prompt "c:\java1>" is visible at the bottom.

Practical 6:

1. Write a Java Program to create threads using Thread class and Runnable Interface.

```
class p61 {  
    public static void main(String[] args) {  
        t1 t1=new t1();  
        t2 t2=new t2();  
        Thread t=new Thread(t2);  
        t1.start();  
        t.start();  
    }  
}  
  
class t1 extends Thread{  
    public void run(){  
        System.out.println("Thread created by extending Thread class");  
    }  
}  
  
class t2 implements Runnable{  
    public void run(){  
        System.out.println("Thread created by implementing Runnable interface");  
    }  
}
```



C:\ Command Prompt

```
c:\java1>java 6p1.java
Thread created by extending Thread class
Thread created by implementing Runnable interface

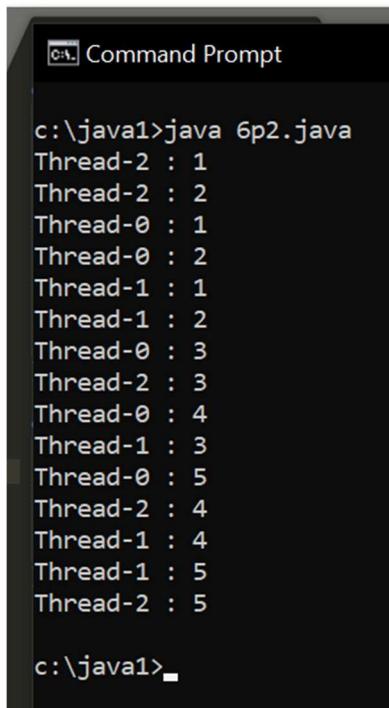
c:\java1>
```

2. Create 3 threads in java and call run to execute a loop. Start all 3 together.

```
class p62 {
    public static void main(String[] args) {
        t t =new t();
        Thread t1=new Thread(t);
        Thread t2=new Thread(t);
        Thread t3=new Thread(t);
        t1.start();
        t2.start();
        t3.start();
    }
}
```

```
class t implements Runnable{
    public void run(){
        for (int i=1;i<=5 ;i++ ) {
```

```
        System.out.println(Thread.currentThread().getName()+" :  
        "+i );  
    }  
}  
}
```



```
c:\java1>java 6p2.java  
Thread-2 : 1  
Thread-2 : 2  
Thread-0 : 1  
Thread-0 : 2  
Thread-1 : 1  
Thread-1 : 2  
Thread-0 : 3  
Thread-2 : 3  
Thread-0 : 4  
Thread-1 : 3  
Thread-0 : 5  
Thread-2 : 4  
Thread-1 : 4  
Thread-1 : 5  
Thread-2 : 5  
c:\java1>
```

3. Show thread life cycle in Java using program. Refer functions – start,run,sleep,notify,notifyAll,wait,stop

```
class p62{  
    public static void main(String[] args) {  
        t t1=new t();  
        t t2=new t();  
        t t3=new t();  
        t1.setName("Thread-A");
```

```
t2.setName("Thread-B");
t3.setName("Thread-C");

System.out.println("Thread-A state-"+ t1.getState());
System.out.println("Thread-B state-"+t2.getState());
System.out.println("Thread-C state-"+t3.getState());

t1.start();
t2.start();
t3.start();

t1.stop();

System.out.println(t1.getName()+" ID "+ t1.getId());
t1.resume();

try{ t3.wait(); }catch(Exception e){ }

System.out.println(t1.getName()+" ID "+ t1.getId());
System.out.println(t2.getName()+" ID "+ t2.getId());
System.out.println(t3.getName()+" ID "+ t3.getId());

System.out.println("currentThread is "+Thread.currentThread());
System.out.println("Thread-A is Alive "+ t1.isAlive());

System.out.println("Thread-A state-"+ t1.getState());
System.out.println("Thread-B state-"+t2.getState());
System.out.println("Thread-C state-"+t3.getState());

}
```

```
}
```

```
class t extends Thread{  
  
    public void run(){  
        System.out.println(Thread.currentThread().getName()+" state "+ getState());  
        System.out.println("creating "+Thread.currentThread().getName());  
        System.out.println(Thread.currentThread().getName()+" is running");  
        System.out.println(Thread.currentThread().getName()+" in sleep");  
        try{ sleep(100); }catch(Exception e){ }  
        System.out.println(Thread.currentThread().getName()+" state "+  
        getState());  
        System.out.println(Thread.currentThread().getName()+" is wake up");  
        System.out.println(Thread.currentThread().getName()+" state "+ getState());  
  
    }  
}
```

```
c:\java1>java 6p3.java
6p3.java:17: warning: [removal] resume() in Thread has been deprecated and marked for removal
    t1.resume();
    ^
Note: 6p3.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
1 warning
Thread-A state-NEW
Thread-B state-NEW
Thread-C state-NEW
Thread-A ID RUNNABLE
Thread-C state RUNNABLE
Thread-B state RUNNABLE
creating Thread-B
creating Thread-C
Thread-C is running
Thread-B is running
Thread-C in sleep
Thread-B in sleep
Thread-A ID 15
Thread-B ID 16
Thread-C ID 17
currentThread is Thread[main,5,main]
Thread-A is Alive false
Thread-A state-TERMINATED
Thread-B state-TIMED_WAITING
Thread-C state-TIMED_WAITING
Thread-B state RUNNABLE
Thread-C state RUNNABLE
Thread-B is wake up
Thread-C is wake up
Thread-B state RUNNABLE
Thread-C state RUNNABLE

c:\java1>
```

4. WAP to run a thread using Join method.

```
class p64{
    public static void main(String[] args) {
        t t1=new t();
        t t2=new t();
        t t3=new t();
```

```
t1.start();
try{ t1.join();
}catch(Exception e){ }

t2.start();
t3.start();

}

class t extends Thread{
    public void run(){
        for (int i=1;i<6 ;i++ ) {

            System.out.println(Thread.currentThread().getName()+" : "+i);
            try{ sleep(100);
}catch(Exception e){ }

        }
    }
}
```

```
c:\java1>java 6p4.java
Thread-0 : 1
Thread-0 : 2
Thread-0 : 3
Thread-0 : 4
Thread-0 : 5
Thread-1 : 1
Thread-2 : 1
Thread-2 : 2
Thread-1 : 2
Thread-2 : 3
Thread-1 : 3
Thread-2 : 4
Thread-1 : 4
Thread-2 : 5
Thread-1 : 5

c:\java1>
```

5. WAP to show use of synchronized method. Create a class with synchronized method. Call it in 3 other thread classes. Start them together.

```
class p65{  
    public static void main(String[] args) {  
        table t=new table();  
        t1 t1 =new t1(t);  
        t2 t2 =new t2(t);  
        t3 t3 =new t3(t);  
  
        t1.start();  
        t2.start();  
        t3.start();  
    }  
}  
  
class table{  
    synchronized void ptable(int n){  
        System.out.println("synchronized methode");  
        for (int i=1;i<11 ;i++ ) {  
            System.out.println(n*i);  
            try{  
                Thread.sleep(100);  
            }  
            catch(Exception e){}  
    }  
}
```

```
        }  
    }  
}  
class t1 extends Thread{  
    table t;  
    t1(table t){  
        this.t=t;  
    }  
    public void run(){  
        t.ptable(1);  
    }  
}  
class t2 extends Thread{  
    table t;  
    t2(table t){  
        this.t=t;  
    }  
    public void run(){  
        t.ptable(2);  
    }  
}  
class t3 extends Thread{  
    table t;  
    t3(table t){  
        this.t=t;  
    }  
    public void run(){
```

```
t.ptable(3);  
}  
}
```

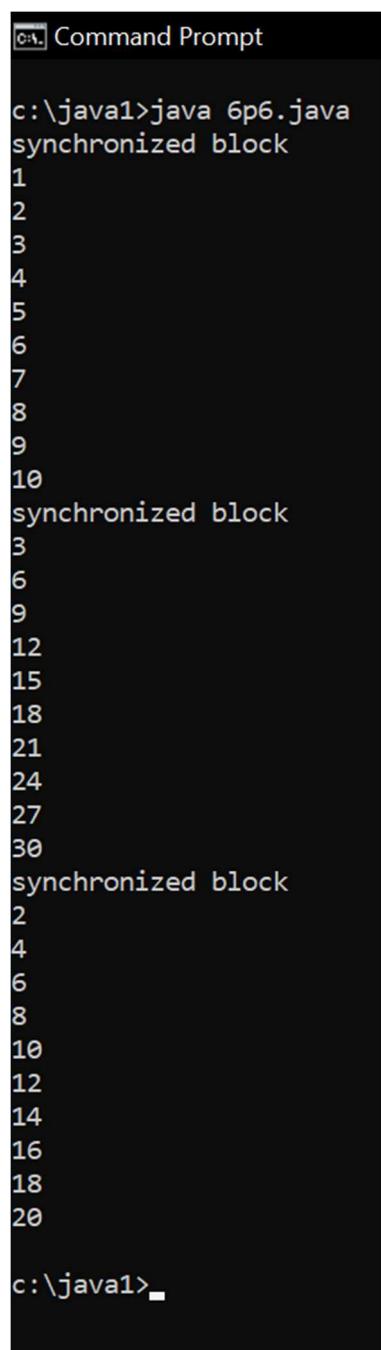
```
c:\java1>java 6p5.java  
synchronized methode  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
synchronized methode  
3  
6  
9  
12  
15  
18  
21  
24  
27  
30  
synchronized methode  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
c:\java1>
```

6. Show the use of synchronized block.

```
class p66{  
    public static void main(String[] args) {  
        table t=new table();  
        t1 t1 =new t1(t);  
        t2 t2 =new t2(t);  
        t3 t3 =new t3(t);  
  
        t1.start();  
        t2.start();  
        t3.start();  
    }  
}  
  
class table{  
    void ptable(int n){  
        synchronized(this){  
            System.out.println("synchronized block");  
            for (int i=1;i<11 ;i++ ) {  
                System.out.println(n*i);  
                try{  
                    Thread.sleep(100);  
                }  
                catch(Exception e){}  
            }  
        }  
    }  
}
```

```
        }  
    }  
}  
  
class t1 extends Thread{  
    table t;  
    t1(table t){  
        this.t=t;  
    }  
    public void run(){  
  
        t.ptable(1);  
    }  
}  
  
class t2 extends Thread{  
    table t;  
    t2(table t){  
        this.t=t;  
    }  
    public void run(){  
        t.ptable(2);  
    }  
}  
  
class t3 extends Thread{  
    table t;  
    t3(table t){  
        this.t=t;  
    }  
}
```

```
public void run(){
    t.ptable(3);
}
```



The screenshot shows a Command Prompt window with the following text:

```
c:\java1>java 6p6.java
synchronized block
1
2
3
4
5
6
7
8
9
10
synchronized block
3
6
9
12
15
18
21
24
27
30
synchronized block
2
4
6
8
10
12
14
16
18
20

c:\java1>
```

7. WAP to show deadlock condition between 3 threads in Java. Then release resource / end one of them to get the normal state

```
//Deadlock condition  
class p67{  
    public static Object Lock1=new Object();  
    public static Object Lock2=new Object();  
    public static Object Lock3=new Object();  
    public static void main(String[] args) {  
        t1 t1=new t1();  
        t2 t2=new t2();  
        t3 t3=new t3();  
        System.out.println(" Deadlock condition ");  
        t1.start();  
        t2.start();  
        t3.start();  
    }  
}
```

```
private static class t1 extends Thread{
```

```
    public void run(){  
        synchronized (Lock1){  
            synchronized (Lock2){  
                synchronized (Lock3){  
                    // code  
                }  
            }  
        }  
    }  
}
```

```
System.out.println(Thread.currentThread().getName()+" is holding  
s1");
```

```
try{sleep(200);}catch(Exception e){}  
System.out.println(Thread.currentThread().getName()+" is waiting  
for s2");
```

```
synchronized (Lock2){  
System.out.println(Thread.currentThread().getName()+" is holding  
s1 and s2 Process complete");  
}  
}  
}
```

```
private static class t2 extends Thread{
```

```
public void run(){  
  
synchronized (Lock2) {  
System.out.println(Thread.currentThread().getName()+" is holding  
s2");  
}  
}
```

```
try{sleep(20);}catch(Exception e){}  
System.out.println(Thread.currentThread().getName()+" is waiting  
for s3");
```

```
synchronized (Lock3){
```

```
        System.out.println(Thread.currentThread().getName()+" is holding  
s3 and s2 Process complete");  
    }  
}  
}  
}  
  
private static class t3 extends Thread{  
  
    public void run(){  
  
        synchronized (Lock3) {  
            System.out.println(Thread.currentThread().getName()+" is holding  
s3");  
  
            try{sleep(20);}catch(Exception e){}  
            System.out.println(Thread.currentThread().getName()+" is waiting  
for s1");  
  
            synchronized (Lock1){  
                System.out.println(Thread.currentThread().getName()+" is holding  
s1 and s3 Process complete");  
            }  
        }  
    }  
}
```

```
//End deadlock
```

```
class p67{  
    public static Object Lock1=new Object();  
    public static Object Lock2=new Object();  
    public static Object Lock3=new Object();  
    public static void main(String[] args) {  
        t1 t1=new t1();  
        t2 t2=new t2();  
        t3 t3=new t3();  
        System.out.println("\n\n Deadlock end by terminating Thread-0 ");  
        t1.start();  
        t2.start();  
        t3.start();  
        t1.stop();  
    }  
}
```

```
private static class t1 extends Thread{
```

```
    public void run(){  
        synchronized (Lock1){  
            System.out.println(Thread.currentThread().getName()+" is holding  
s1");  
        }  
    }  
}
```

```
try{sleep(20);}catch(Exception e){}
    System.out.println(Thread.currentThread().getName()+" is waiting
for s2");

    synchronized (Lock2){
        System.out.println(Thread.currentThread().getName()+" is holding
s1 and s2 Process complete");
    }
}

private static class t2 extends Thread{

    public void run(){

        synchronized (Lock2) {
            System.out.println(Thread.currentThread().getName()+" is holding
s2");
        }

        try{sleep(20);}catch(Exception e){}
            System.out.println(Thread.currentThread().getName()+" is waiting
for s3");
        }

        synchronized (Lock3){
            System.out.println(Thread.currentThread().getName()+" is holding
s3 and s2 Process complete");
        }
    }
}
```

```
        }  
    }  
}  
  
private static class t3 extends Thread{  
  
    public void run(){  
  
        synchronized (Lock3) {  
            System.out.println(Thread.currentThread().getName()+" is holding  
s3");  
  
            try{sleep(20);}catch(Exception e){}  
            System.out.println(Thread.currentThread().getName()+" is waiting  
for s1");  
  
            synchronized (Lock1){  
                System.out.println(Thread.currentThread().getName()+" is holding  
s1 and s3 Process complete");  
            }  
        }  
    }  
}
```

```
c:\java1>java 6p7.java
Deadlock condition
Thread-2 is holding s3
Thread-1 is holding s2
Thread-0 is holding s1
Thread-1 is waiting for s3
Thread-2 is waiting for s1
Thread-0 is waiting for s2
p^C
c:\java1>java 6p7-1.java
Note: 6p7-1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

Deadlock end by terminating Thread-0
Thread-2 is holding s3
Thread-1 is holding s2
Thread-1 is waiting for s3
Thread-2 is waiting for s1
Thread-2 is holding s1 and s3 Process complete
Thread-1 is holding s3 and s2 Process complete

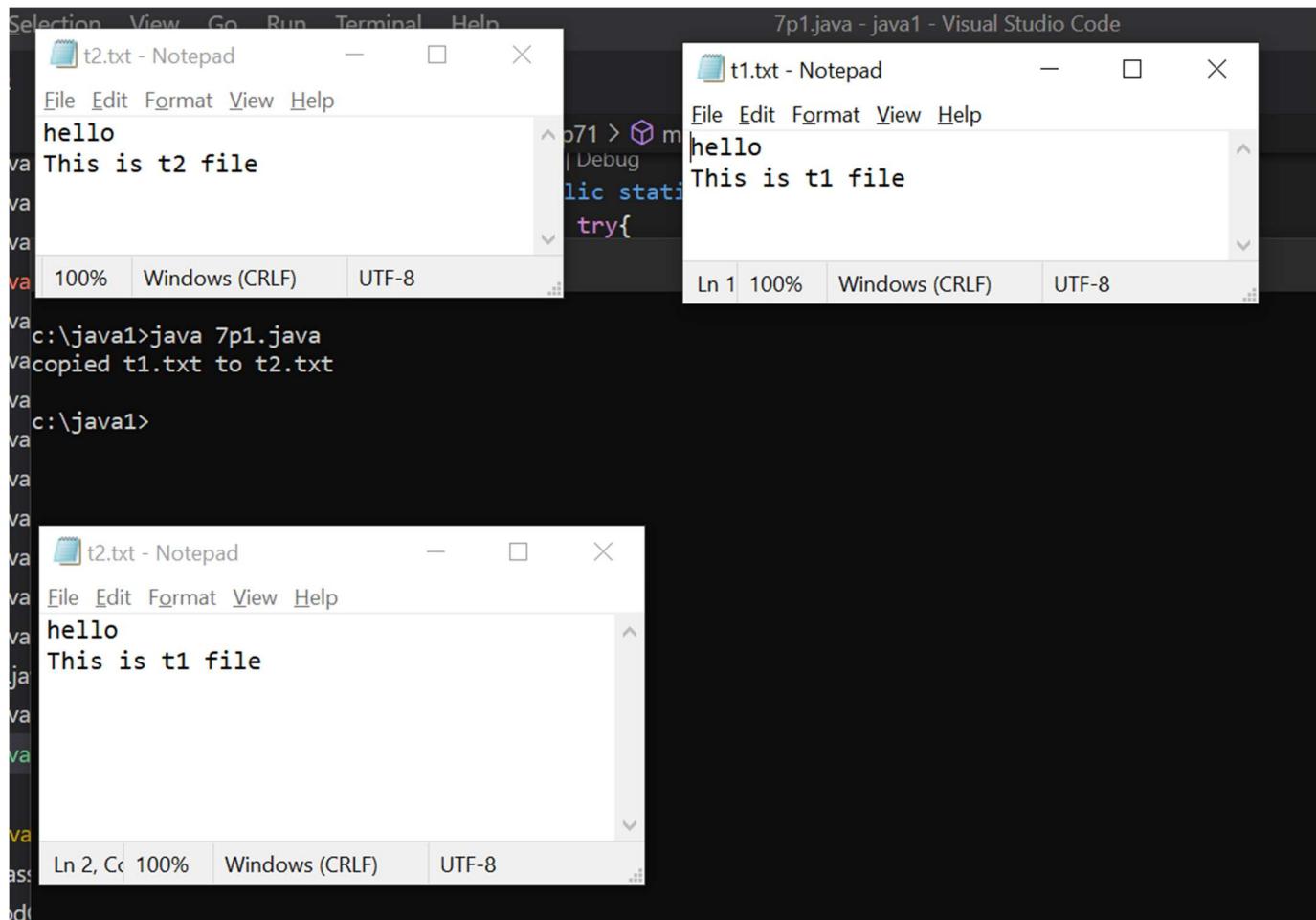
}c:\java1>_
```

Practical 7:

1. Write a Java Program to copy contents from one file into another using FileInputStream & FileOutputStream.

```
import java.io.*;
class p71{
    public static void main(String[] args) {
        try{
            File in=new File("D:\\html\\New\\1.txt");
            File out=new File("D:\\html\\New\\2.txt");
            FileInputStream fin=new FileInputStream(in);
            FileOutputStream fout=new FileOutputStream(out);
            byte[] b=new byte[1024];
            int l;
            while ((l=fin.read(b))>0) {
                fout.write(b,0,l);

            }
            fin.close();
            fout.close();
            System.out.println("copied 1.jpg to 2.jpg");
        }catch(IOException e){ e.printStackTrace();}
    }
}
```

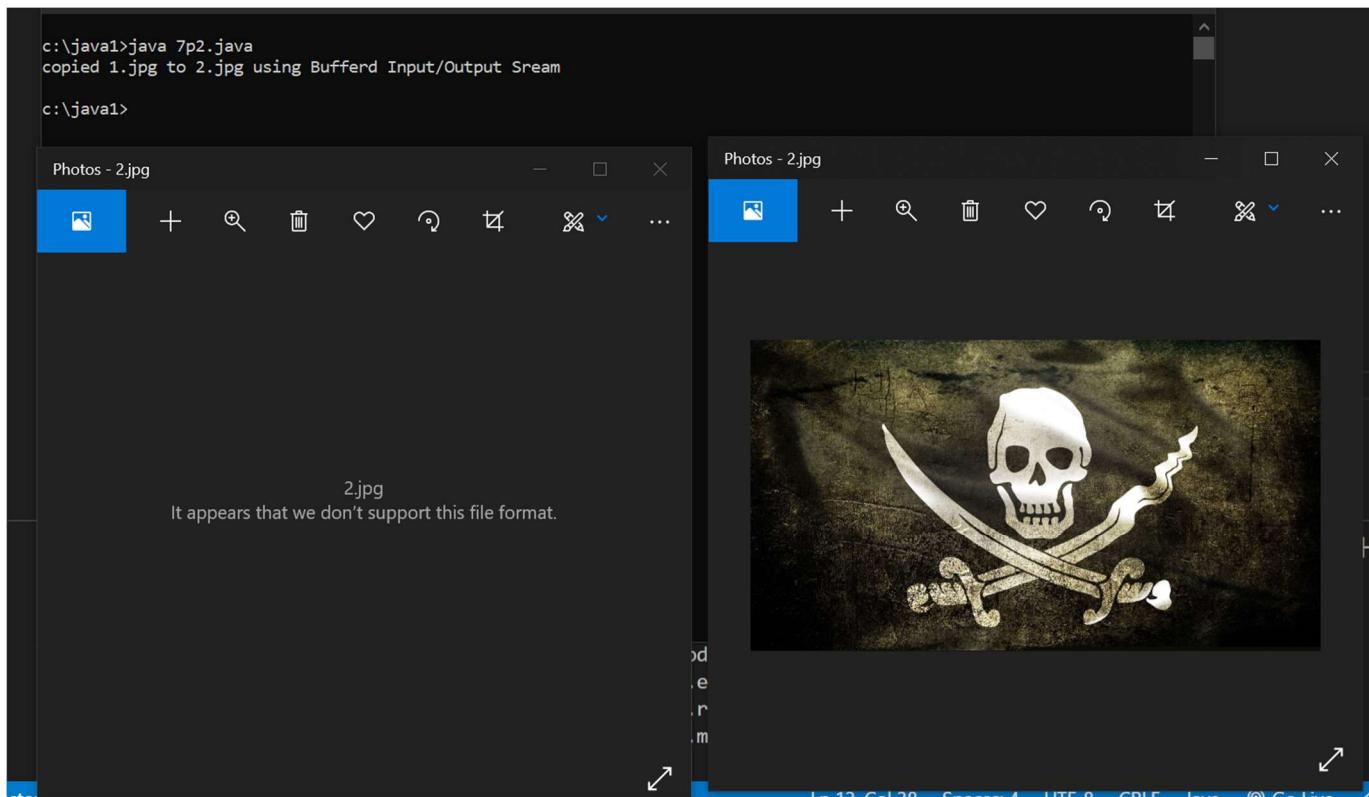


2. Write a Java Program to copy contents from one file into another using BufferedInputStream & BufferedOutputStream.

```
import java.io.*;
public class p2 {
    public static void main(String[] args) {
        File in=new File("D:\\html\\New\\1.jpg");
        File out=new File("D:\\html\\New\\2.jpg");
        try{
            FileInputStream fin=new FileInputStream(in);
```

```
FileOutputStream fout=new FileOutputStream(out);
BufferedInputStream bin=new BufferedInputStream(fin);
BufferedOutputStream bout=new BufferedOutputStream(fout);
byte[] b=new byte[1024];
int l;
while ((l=bin.read(b))!=-1) {
    bout.write(b,0,l);
}

try{fin.close();
    fout.close();
    bin.close();
    bout.close();
}catch(IOException e){}
System.out.println("copied 1.jpg to 2.jpg using Bufferd Input/Output
Stream");
}catch(IOException e){ e.printStackTrace();}
}
```



3. WAP to copy contents from a file and arrange the words in ascending order to store in another file using Reader and Writer classes

```
import java.io.*;
import java.util.*;
import java.nio.CharBuffer;
import java.util.Arrays;
public class p3 {
    public static void main(String[] args) {
        int i,j=0;
        char []a=new char[30];
        try{
```

```
FileReader fr=new FileReader("D:/html/t.txt");
FileWriter fw= new FileWriter("D:/html/o.txt");
while((i=fr.read())!=-1){

    a[j]=(char)i;
    j++;
}

Arrays.sort(a);
fw.write(a);
//System.out.println("a=%s"+Arrays.toString(a));
System.out.println("sorting done t.txt to o.txt");
fr.close();
fw.close();
}catch(Exception e){}
}

}
```

c:\java1>java 7p3.java
sorting done t.txt to o.txt
c:\java1>

The screenshot shows a terminal window at the top with the command `c:\java1>java 7p3.java` and the output "sorting done t.txt to o.txt". Below the terminal are two Notepad windows. The left window, titled "t.txt - Notepad", contains the乱码 text "qwertyuiopasdfghjklmnbcxz". The right window, titled "o.txt - Notepad", contains the sorted text "abcdefghijklmnopqrstuvwxyz".

4. WAP in Java to use map and store username and password values in properties. Also, retrieve them in program when asked.

```
import java.util.*;  
import java.io.*;  
public class p744{  
    public static void main(String[] args) {  
        Properties p=new Properties();  
        Map<String, String> hm=new HashMap<String, String>();  
        hm.put("@aftab", "Aal");
```

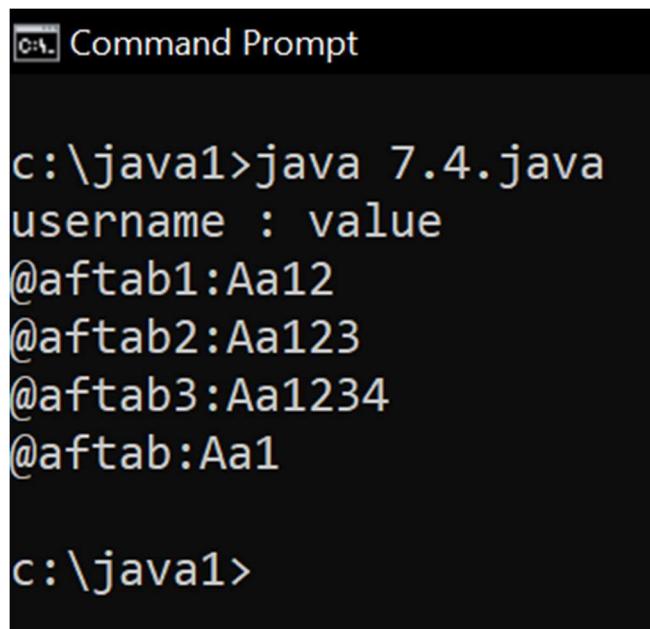
```
hm.put("@aftab1", "Aa12");
hm.put("@aftab2", "Aa123");
hm.put("@aftab3", "Aa1234");
for(Map.Entry<String, String> me : hm.entrySet())
{
    p.setProperty(me.getKey(), me.getValue());
    // System.out.println(me.getKey()+":"+me.getValue());
}
try{

    p.store(new FileWriter("D:/html/p.txt"), "usernames and passwords");
} catch(Exception e){}
System.out.println("username : value");

// System.out.println(p.toString());
Set set=p.entrySet();
Iterator itr=set.iterator();
while (itr.hasNext()) {
    Map.Entry entry=(Map.Entry)itr.next();
    System.out.println(entry.getKey()+":"+entry.getValue());
}

// for(Map.Entry<String, String> me : hm.entrySet())
// {
//     // System.out.println(hm);
//     System.out.println(me.getKey()+":"+me.getValue());
// }
}
```

```
}
```



```
c:\java1>java 7.4.java
username : value
@aftab1:Aa12
@aftab2:Aa123
@aftab3:Aa1234
@aftab:Aa1

c:\java1>
```

5. WAP in Java to implement HashMap and perform operations to insert if not present, replace & delete data. Also, iterate over each pair.

```
import java.util.*;

public class p755 {
    public static void main(String[] args) {
        Map<String, String> hm=new HashMap<String, String>();
        hm.put("@aftab", "Aa1");
        hm.put("@aftab1", "Aa12");
        hm.put("@aftab2", "Aa123");
        hm.put("@aftab3", "Aa1234");
        System.out.println("Map=");
    }
}
```

```
for(Map.Entry<String, String> me : hm.entrySet())
{
    // System.out.println(hm);
    System.out.println(me.getKey()+":"+me.getValue());
}

hm.put("@aftab1", "changed");
hm.remove("@aftab3");
System.out.println("Map after removing @aftab3 and changing @aftab1 =");
for(Map.Entry<String, String> me : hm.entrySet())
{
    // System.out.println(hm);
    System.out.println(me.getKey()+":"+me.getValue());
}
}
```



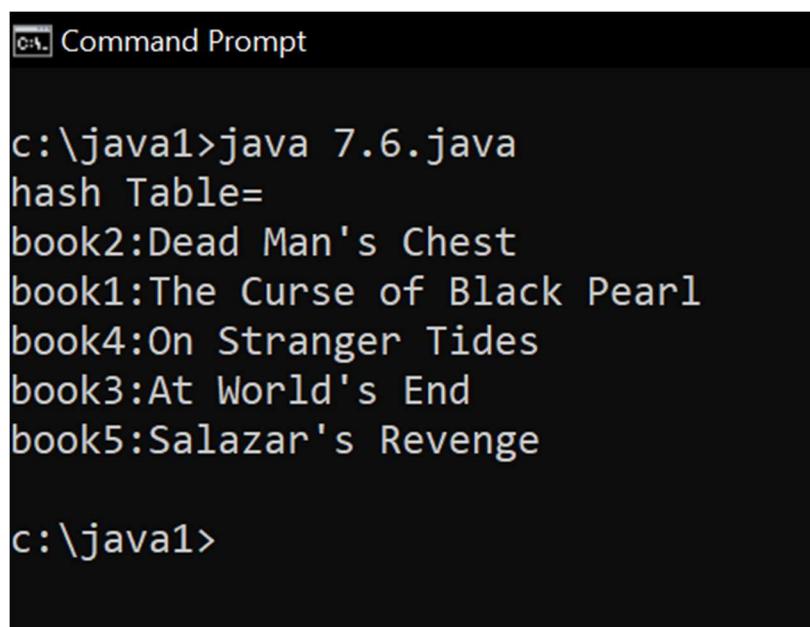
The screenshot shows a Command Prompt window with the following text:

```
c:\java1>java 7.5.java
Map=
@aftab1:Aa12
@aftab2:Aa123
@aftab3:Aa1234
@aftab:Aa1
Map after removing @aftab3 and changing @aftab1 =
@aftab1:changed
@aftab2:Aa123
@aftab:Aa1
c:\java1>
```

6. WAP in Java to implement HashTable to store details of books in a library & retrieve them

```
import java.util.*;
import java.io.*;
public class p766 {
    public static void main(String[] args) {
        Hashtable<String, String> ht=new Hashtable<String, String>();
        Properties p=new Properties();
        ht.put("book1", "The Curse of Black Pearl");
        ht.put("book2", "Dead Man's Chest");
        ht.put("book3", "At World's End");
        ht.put("book4", "On Stranger Tides");
        ht.put("book5", "Salazar's Revenge");
        for(Map.Entry<String, String> me : ht.entrySet())
        {
            p.setProperty(me.getKey(), me.getValue());
            // System.out.println(me.getKey()+":"+me.getValue());
        }
        try{
            p.store(new FileWriter("D:/html/library.txt"), "library");
        }catch(Exception e){}
        System.out.println("hash Table=");
        // for(Map.Entry<String, String> me : ht.entrySet())
```

```
// {  
//     System.out.println(me.getKey()+":"+me.getValue());  
// }  
  
Set set=p.entrySet();  
Iterator itr=set.iterator();  
while (itr.hasNext()) {  
    Map.Entry entry=(Map.Entry)itr.next();  
    System.out.println(entry.getKey()+":"+entry.getValue());  
}  
}  
}
```



```
c:\java1>java 7.6.java  
hash Table=  
book2:Dead Man's Chest  
book1:The Curse of Black Pearl  
book4:On Stranger Tides  
book3:At World's End  
book5:Salazar's Revenge  
  
c:\java1>
```

Practical 8:

1. WAP in Java to implement one-way TCP based client server communication in Java

Client side :

```
// Client to server one way communication
import java.io.*;
import java.net.*;
// class MyClient {
// public static void main(String[] args) {
// try{
// Socket s=new Socket("localhost",6666);
// DataOutputStream dout=new DataOutputStream(s.getOutputStream());
// dout.writeUTF("Hello Server");
// dout.flush();
// dout.close();
// s.close();
// }catch(Exception e){System.out.println(e);}
// }
// }

//server to client one way communication
class Client
{
    public static void main(String args[]) throws Exception
    {
        Socket sock = new Socket("127.0.0.1", 7000);

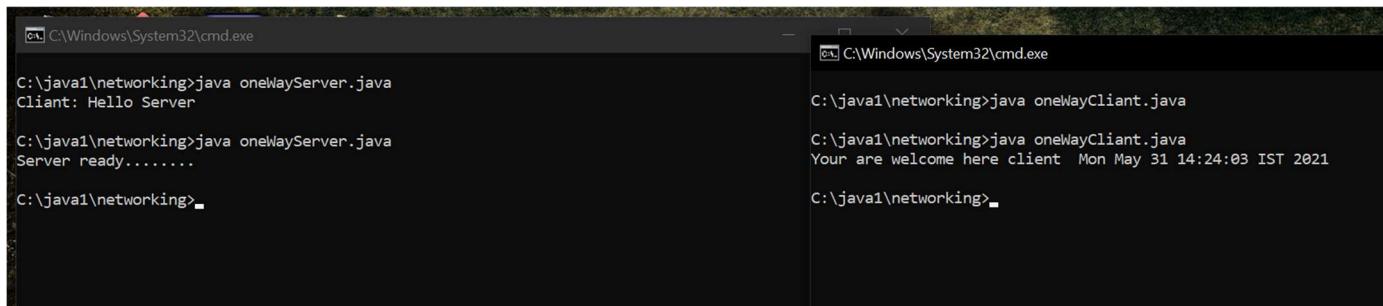
        InputStream istream = sock.getInputStream();
```

```
BufferedReader br1 = new BufferedReader(new InputStreamReader(istream));  
  
String s1 = br1.readLine();  
System.out.println(s1);  
  
br1.close(); istream.close(); sock.close();  
}  
}
```

Server side :

```
// Client to server one way communication  
  
import java.io.*;  
import java.net.*;  
  
// class MyServer{  
  
// public static void main(String[] args){  
  
// try{  
  
// ServerSocket ss=new ServerSocket(6666);  
  
// Socket s=ss.accept();//establishes connection  
  
// DataInputStream dis=new DataInputStream(s.getInputStream());  
  
// String str=(String)dis.readUTF();  
  
// System.out.println("Client: "+str);  
  
// ss.close();  
  
// }catch(Exception e){System.out.println(e);}  
// }  
// }  
  
//server to client one way communication  
  
class DateInfoServer
```

```
{  
    public static void main(String args[]) throws Exception  
{  
        ServerSocket sersock = new ServerSocket(7000);  
        System.out.println("Server ready.....");  
        Socket sock = sersock.accept();  
  
        OutputStream ostream = sock.getOutputStream();  
        BufferedWriter bw1 = new BufferedWriter(new OutputStreamWriter(ostream));  
        String s2 = "Your are welcome here client " + new java.util.Date();  
        bw1.write(s2);  
  
        bw1.close();  ostream.close();  sock.close();  sersock.close();  
    }  
}
```



2. WAP in Java to implement two-way TCP based client server communication in Java

Server side:

```
import java.net.*;  
import java.io.*;
```

```
class MyServer{
public static void main(String args[])throws Exception{
    try{
        ServerSocket ss=new ServerSocket(3333);
        Socket s=ss.accept();
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        String str="",str2="";
        while(!str.equals("bey")){
            str=din.readUTF();
            System.out.println("client: "+str);
            str2=br.readLine();
            dout.writeUTF(str2);
            dout.flush();
        }
        din.close();
        s.close();
        ss.close();
    }catch(Exception e){}
}}
```

Client side:

```
import java.net.*;
import java.io.*;
class MyClient{
public static void main(String args[])throws Exception{
    try{
        Socket s=new Socket("localhost",3333);
```

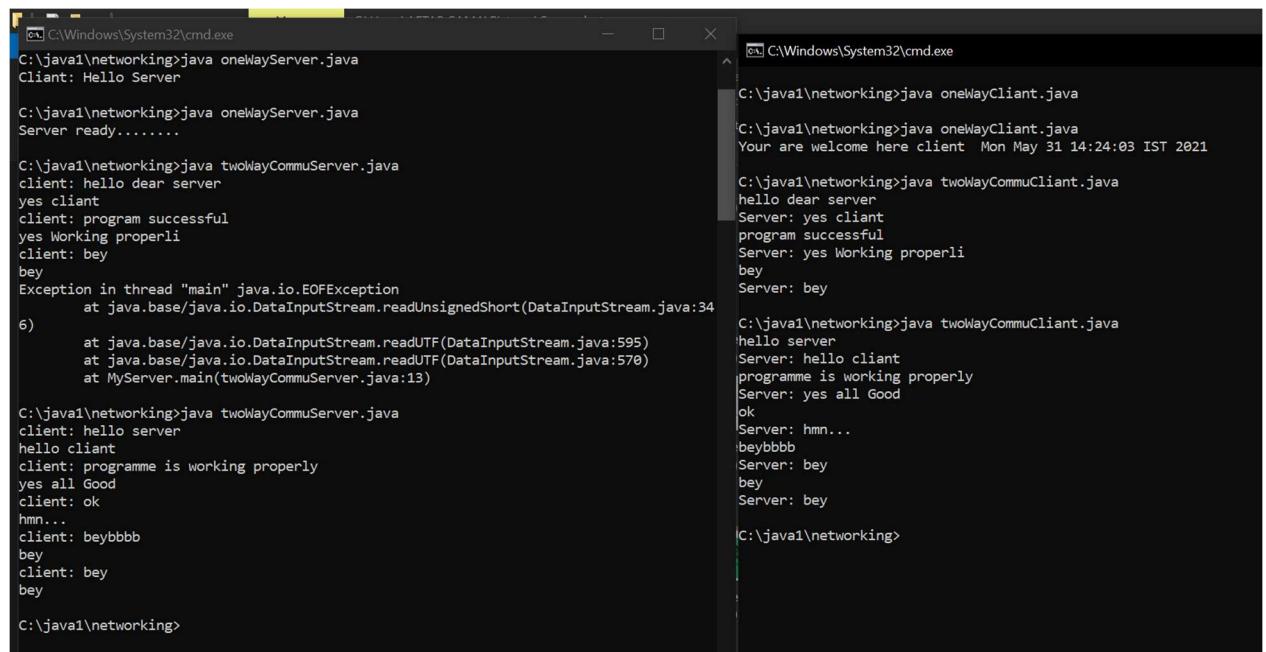
```

DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("bey")){
    str=br.readLine();
    dout.writeUTF(str);
    dout.flush();
    str2=din.readUTF();
    System.out.println("Server: "+str2);
}

dout.close();
s.close();
}catch(Exception e){}
}}

```



The image shows two terminal windows side-by-side, both titled 'C:\Windows\System32\cmd.exe'. The left window contains the output of a Java application named 'oneWayServer.java'. It starts with 'Client: Hello Server' and then 'Server ready.....'. It then receives input from another client, which includes a stack trace for an EOFException. The right window contains the output of a Java application named 'oneWayClient.java'. It starts with 'Your are welcome here client Mon May 31 14:24:03 IST 2021'. It then sends a message to the server, which replies with 'hello dear server', 'Server: yes client', and 'program successful'. The client then sends 'bey', and the server replies with 'bey' again.

```

C:\java1\networking>java oneWayServer.java
Client: Hello Server
C:\java1\networking>java oneWayServer.java
Server ready.....
C:\java1\networking>java twoWayCommuServer.java
client: hello dear server
yes client
client: program successful
yes Working properli
client: bey
bey
Exception in thread "main" java.io.EOFException
    at java.base/java.io.DataInputStream.readUnsignedShort(DataInputStream.java:346)
    at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:595)
    at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:570)
    at MyServer.main(twoWayCommuServer.java:13)

C:\java1\networking>java twoWayCommuServer.java
client: hello server
hello client
client: programme is working properly
yes all Good
client: ok
hm...
client: beybbbb
bey
client: bey
bey

C:\java1\networking>

```

```

C:\Windows\System32\cmd.exe
C:\java1\networking>java oneWayClient.java
C:\java1\networking>java oneWayClient.java
Your are welcome here client Mon May 31 14:24:03 IST 2021
C:\java1\networking>java twoWayCommuClient.java
hello dear server
Server: yes client
program successful
Server: yes Working properli
bey
Server: bey

C:\java1\networking>java twoWayCommuClient.java
hello server
Server: hello client
programme is working properly
Server: yes all Good
ok
Server: hm...
beybbbb
Server: bey
bey
Server: bey

C:\java1\networking>

```

3. WAP to implement UDP based client server communication in Java

Sender:

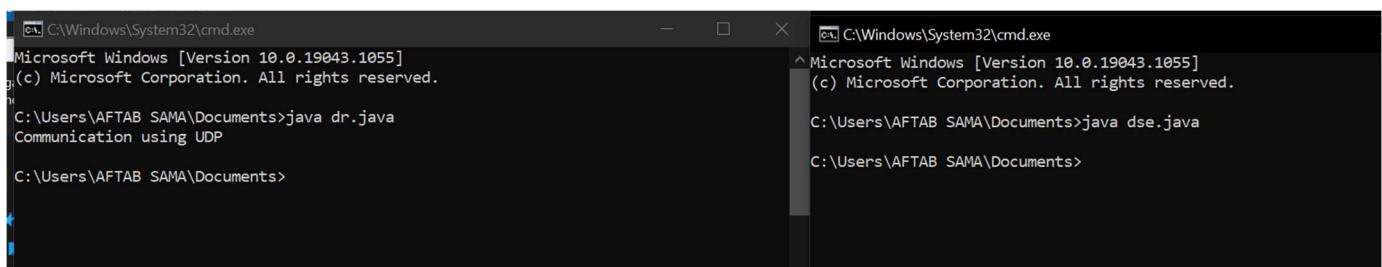
```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.nio.charset.StandardCharsets;

public class dse {
    public static void main(String[] args)
    {
        try {
            DatagramSocket ds = new DatagramSocket();
            String s="Communication using UDP";
            DatagramPacket dp=new
DatagramPacket(s.getBytes(StandardCharsets.UTF_8),s.length(),InetAddress.getByName("lo
calhost"),6677);
            ds.send(dp);
            ds.close();
        }
        catch (Exception e)
        {
            }
    }
}
```

resever:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class dr {
    public static void main(String[] args)
    {
        try{
            DatagramSocket ds=new DatagramSocket(6677);
            byte[] buf=new byte[512];
            DatagramPacket dp=new DatagramPacket(buf,512);
            ds.receive(dp);
            System.out.println(new String(dp.getData(),0,dp.getLength()));
            ds.close();
        }
        catch (Exception e)
        {
        }
    }
}
```



4. WAP to create Chat application in Java.

Server:

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;
import java.security.KeyStore;

public class chserver2 {
    public static void main(String[] args)
    {
        int i;
        try{
            ServerSocket ss=new ServerSocket(6677);
            Socket s=ss.accept();
            send t1=new send(s);
            receive t2=new receive(s);

            t1.start();t2.start();
        }
        catch (Exception e)
        {
    }
}
```

```
    }

}

class send extends Thread{

    Socket s;
    String str2="";
    public send(Socket s)
    {
        this.s=s;
    }
    public void run()
    {
        try{
            while (!str2.equals("byy")){
                DataOutputStream dout = new DataOutputStream(s.getOutputStream());
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                str2 = br.readLine();
                dout.writeUTF(str2);
                dout.flush();
            }
        }catch (Exception e){
            {
        }
    }
}

class receive extends Thread
{
```

```
Socket s;
String str1="";
public receive(Socket s)
{
    this.s=s;
}
public void run()
{
    try {
        while (!str1.equals("byy")) {
            DataInputStream din = new DataInputStream(s.getInputStream());
            str1 = din.readUTF();
            System.out.println("client say:" + str1);
        }
    } catch (Exception e) {
    }
}

}
```

Client:

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.InetAddress;
```

```
import java.net.Socket;

public class chclient2{
    public static void main(String[] args)
    {
        try {
            Socket s = new Socket("localhost", 6677);

            String str1="";
            String str2="";

            se t1=new se(s);
            re t2=new re(s);
            t1.start();t2.start();

        }
        catch (Exception e)
        {

        }
    }
}
```

```
class se extends Thread{  
    Socket s;  
    String str2="";  
    public se(Socket s)  
    {  
        this.s=s;  
    }  
    public void run()  
    {  
        try{  
            while(!str2.equals("byy")) {  
                DataOutputStream dout = new DataOutputStream(s.getOutputStream());  
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
                str2 = br.readLine();  
                dout.writeUTF(str2);  
                dout.flush();  
            }  
        }  
        catch (Exception e)  
        {  
        }  
    }  
}  
class re extends Thread  
{  
    Socket s;  
    String str1="";  
    public re(Socket s)
```

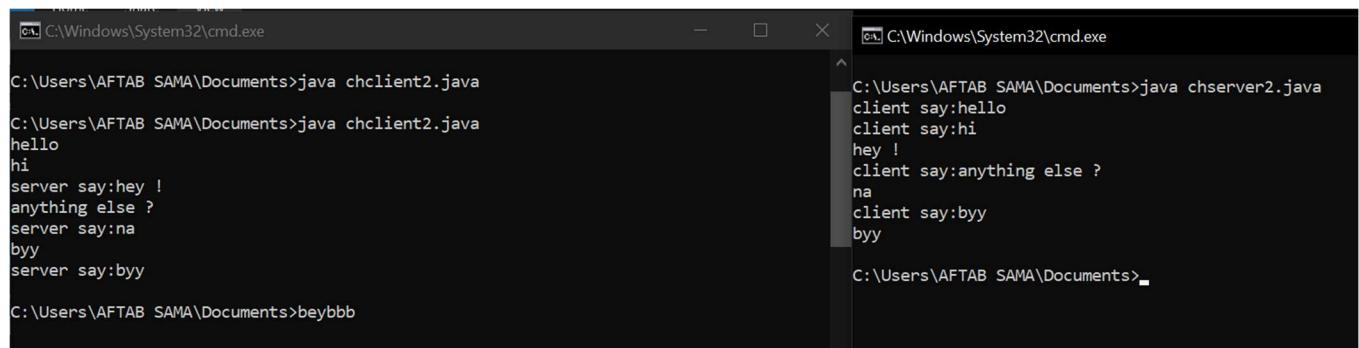
```

{
    this.s=s;
}
public void run()
{
try {
    while (!str1.equals("byy")) {

        DataInputStream din = new DataInputStream(s.getInputStream());
        str1 = din.readUTF();
        System.out.println("server say:" + str1);
    }
}
catch (Exception e)
{
}

}
}

```



The screenshot shows two separate command-line windows. The left window (cmd.exe) represents the client side, and the right window represents the server side. Both windows are running under the command prompt on Windows.

Client Side (Left Window):

```

C:\Users\AFTAB SAMA\Documents>java chclient2.java
C:\Users\AFTAB SAMA\Documents>java chclient2.java
hello
hi
server say:hey !
anything else ?
server say:na
byy
server say:byy

C:\Users\AFTAB SAMA\Documents>beybbb

```

Server Side (Right Window):

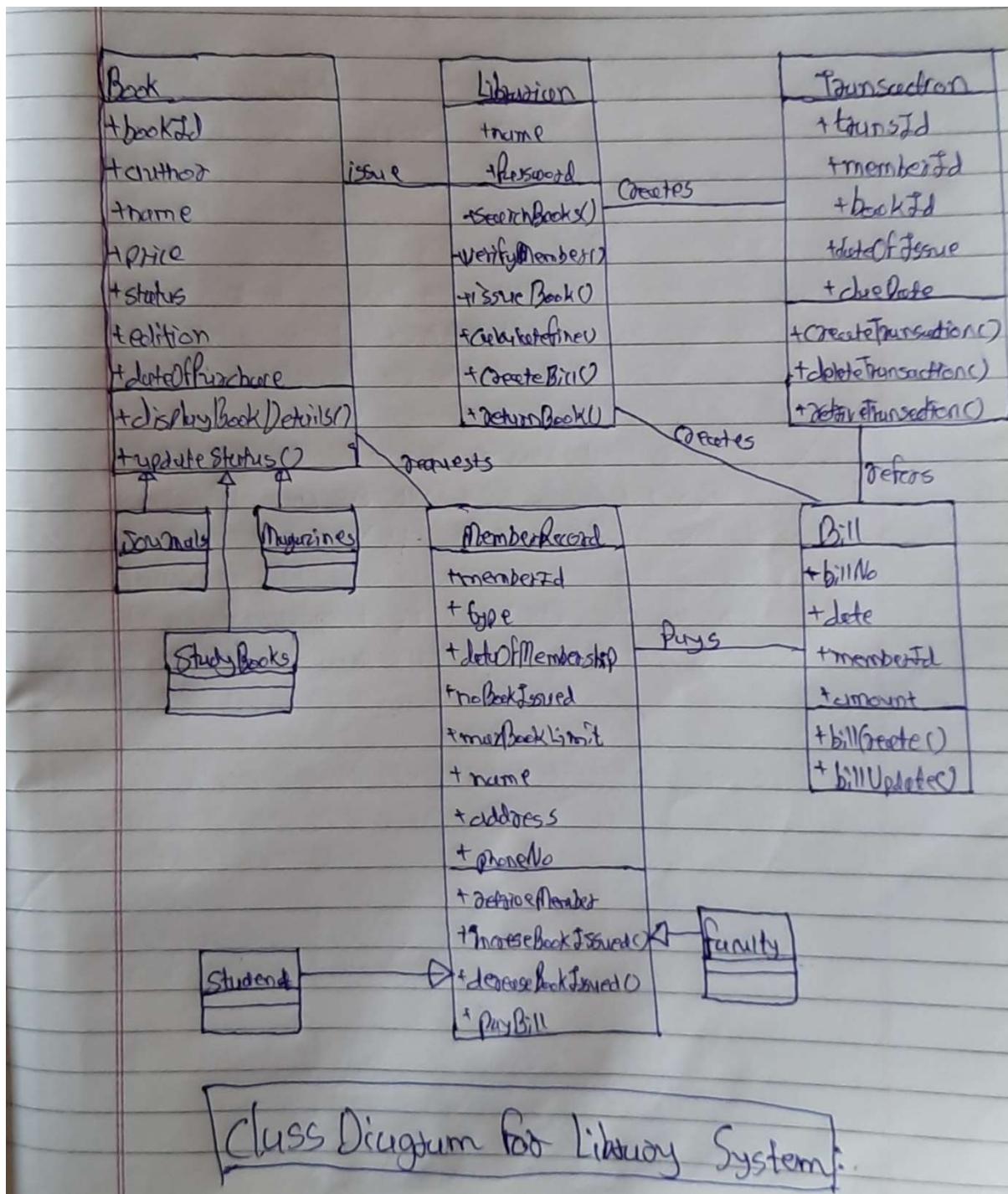
```

C:\Users\AFTAB SAMA\Documents>java chserver2.java
client say:hello
client say:hi
hey !
client say:anything else ?
na
client say:byy
byy

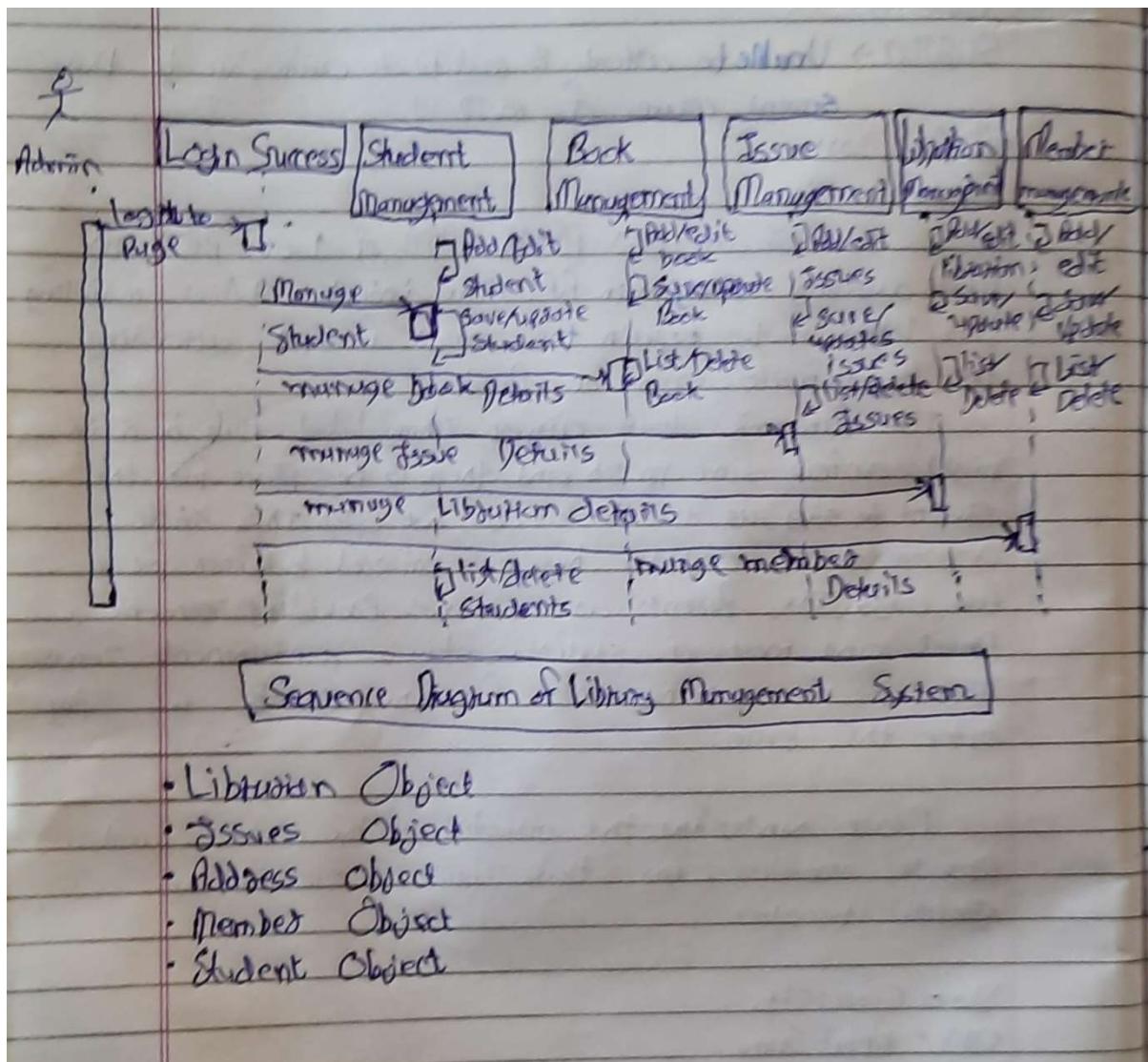
C:\Users\AFTAB SAMA\Documents>

```

5. Depict Advanced Class modelling diagram for any one of the following management systems: Bank / Library/ Hostel/ Student / Employee.



6. Depict Sequence modelling diagram for any one of the following management systems: Bank / Library/ Hostel/ Student / Employee.



CHALLENGE PROGRAMS:

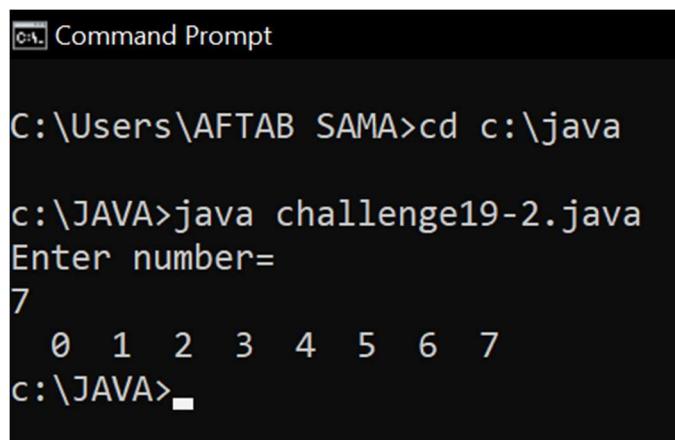
1. Recursively print nos. from 0 to n such that only one parameter is passed to recursive function.

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

```
public class Main {
```

```
    public static int karo(int n) {
```

```
int i;  
for(i=0;i<=n;i++){  
    System.out.print(" "+i);  
}  
return n;  
}  
  
public static void main(String[] args) {  
  
    Scanner sc= new Scanner(System.in);  
    System.out.println("Enter number=");  
    int n= sc.nextInt();  
    karo( n);  
}  
}
```



```
C:\Users\AFTAB SAMA>cd c:\java  
c:\JAVA>java challenge19-2.java  
Enter number=  
7  
0 1 2 3 4 5 6 7  
c:\JAVA>
```

2. WAP to recursively remove repeating characters in a string given by user.

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

```

class cp2{
    public static String rr(String input) {
        if(input.length()<=1)
            return input;
        if(input.charAt(0)==input.charAt(1))
            return rr(input.substring(1));
        else
            return input.charAt(0) + rr(input.substring(1));
    }
    public static void main(String[] args)
    {
        String S1 = "aafftabb";
        System.out.println(rr(S1));

        String S2 = "ssamma";
        System.out.println(rr(S2));
    }
}

```

```

C:\ Command Prompt
c:\JAVA>java cp2.java
aftab
sama
c:\JAVA>

```

3. Starting with 1st Jan 2000, any other date when entered should give day.

```

import java.util.*;
class day{
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int i,j,n,d,m,y;
        int c=2020,day;

```

```
int x[]={};  
System.out.println("Day=");  
d=sc.nextInt();  
System.out.println("Month=");  
m=sc.nextInt();  
System.out.println("Year=");  
y=sc.nextInt();  
  
x[0]=4;  
x[1]=0;  
x[2]=0;  
x[3]=3;  
x[4]=5;  
x[5]=1;  
x[6]=3;  
x[7]=6;  
x[8]=2;  
x[9]=4;  
x[10]=0;  
x[11]=2;  
  
if (y>2021) {  
    for (i=0;y>2021 ;i++ ) {  
        for (j=0;j<12 ;j++ ) {  
            x[j]++;  
        }  
        if (y%4==0) {  
            for (j=0;j<12 ;j++ ) {  
                x[j]++;  
            }  
        }  
    }  
}
```

```
for (j=2;j<12 ;j++ ) {  
    x[j]++;  
}  
}  
if ((y-1)%4==0) {  
    for (j=0;j<2;j++ ) {  
        x[j]++;  
    }  
}  
y--;  
}  
}  
if (y<2021){  
    for(;y<2021;){  
        for (n=0;n<12 ;n++ ) {  
            if(x[n]==0) { x[n]=7; } x[n]--;  
        }  
        if (y % 4 ==0) { for ( i=0;i<2 ;i++ ) {  
            if(x[i]==0) { x[i]=7; } x[i]--;  
        }  
        if ((y+1) % 4 == 0) { for (n=2;n<12 ;n++ ) {  
            if(x[n]==0) { x[n]=7; } x[n]--;  
        }  
    }  
}
```

```
}
```

```
}
```

```
y++;
```

```
}
```

```
}
```

```
day=d+x[m-1];
for (i=0;day>7 ; ) {
    day=day-7;
}
if (day==1) {
    System.out.println("Day is Monday");
}
if (day==2) {
    System.out.println("Day is Tuesday");
}
if (day==3) {
    System.out.println("Day is Wednesday");
}
if (day==4) {
    System.out.println("Day is Thursday");
```

```
}

if (day==5) {
    System.out.println("Day is Friday");
}

if (day==6) {
    System.out.println("Day is Saturday");
}

if (day==7) {
    System.out.println("Day is Sunay");
}

}
```

```
c:\ Command Prompt

c:\java1>java d2d.java
Day=
13
Month=
2
Year=
2020
Day is Thursday

c:\java1>java d2d.java
Day=
25
Month=
3
Year=
2019
Day is Monday

c:\java1>java d2d.java
Day=
23
Month=
10
Year=
2024
Day is Wednesday

c:\java1>java d2d.java
Day=
14
Month=
3
Year=
2025
Day is Friday

c:\java1>
```

4. WAP in Java to create custom exception, call the exception, rethrow it and handle the exception.

```
class TestCustomException1 {  
  
    static void validate(int age) throws InvalidAgeException {  
        if (age < 18)  
            throw new InvalidAgeException("not valid");  
        else  
            System.out.println("welcome to vote");  
    }  
  
    public static void main(String args[]) {  
        try {  
            validate(13);  
        } catch (Exception m) { System.out.println("Exception occurred: "+m); }  
    }  
}  
  
class InvalidAgeException extends Exception {  
    InvalidAgeException(String s) {  
        super(s);  
    }  
}
```

```
c:\ Command Prompt
c:\JAVA>java cp4.java
Exception occurred: InvalidAgeException: not valid

c:\JAVA>
```

*

The End