1). Write a program in Java, to implement the stack data structure.

```
class stack {
    private int arr[];
    private int top;
    private int capacity;

    stack (int size) {
        arr = new int[size];
        capacity = size;
        top = -1;
    }
    public void push (int x) {
        if (isFull()) {
            System.out.println("Stack overflow");
            System.exit(1);
        }
        System.out.println("Inserting "+x);
        arr[++top] = x;
    }
    public int pop() {
        if (isEmpty()) {
            System.out.println("Stack Empty");
            System.exit(1);
        }
        return arr[top--];
    }
```

```
public int getSize() { return top+1; }
public boolean isEmpty() { return top == -1; }
public boolean isFull() { return top == capacity-1; }
public void printStack() {
    for ( int i=0 ; i <= top ; i++) {
        System.out.print ( arr[i] + ", ");
    }
}
public static void main ( String args[]) {
    stack Stack = new stack (5);
    Stack.push (1);
    Stack.push (2);
    Stack.push (3);
    System.out.print (" Stack: ");
    stack.printStack();
    Stack.pop();
    System.out.println ("\n After popping out ");
    Stack.printstack();
    Stack.pop();
    System.out.println (" After popping out ");
    Stack.printstack();
  }
}
```

Operations performed:

→ Inserting 1

→ Inserting 2

→ Inserting 3

→ pop()

→ pop()

Output:

Stack: 1, 2, 3,

After popping out

1, 2,

After popping out

1,

2). Write a program in java to implement a simple bank account.

```
import java.util.Scanner;
class BankDetails {
        private string accno;
        private string name;
        private string acc_type;
        private long balance;
        Scanner sc = new Scanner (System.in);

    public void openAccount() {
            System.out.print ("Enter account no:");
            accno = sc.next();
            System.out.print("Enter account type:");
            acc_type = sc.next();
            System.out.print ("Enter Name :");
            name = sc.next();
            System.out.print ("Enter balance:");
            balance = sc.nextLong();
    }
    public void showAccount() {
            System.out.println ("Name of account holder:" +
                                                name);
            System.out.println ("Account no.:" + accno);
            System.out.println ("Account type:" + acc_type);
            System.out.println ("Balance :" + balance);
    }
```

```
public void deposit () {
        long amt ;
        System. out. println ("Enter the amount you
                                want to deposit : ");
        amt = sc. nextLong ();
        balance = balance + amt ;
}
public void withdrawal() {
        long amt ;
        System. out. println ("Enter the amount you
                                want to withdraw:");
        amt = sc. nextLong ();
        if ( balance >= amt) {
                balance = balance - amt ;
                System. out. println ("Balance after
                                withdrawal :" + balance );
        } else {
                System. out. println ("Your balance is less
                        than " + amt + "\t Transition
                                failed... !! ");
        }
}
public boolean search ( String ac_no) {
        if ( accno. equals (ac_no) {
                showAccount ();
                return true ;
        }
        return false ;
```

```
    }
}
public class BankingApp {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("How many number of
                                customer do you want");
        int n = sc.nextInt(); int ch;
        BankDetails c[] = new BankDetails[n];
        for(int i=0 ; i< c.length ; i++) {
            c[i] = new BankDetails();
            c[i].openAccount();
        }
        do {
            System.out.println("\n BANKING SECTION ");
            System.out.println("1. Display all account details
                                \n 2. Search by account number
                                \n 3. Search by CIF.
                                \n 4. Deposit
                                \n 5. Withdrawal
                                \n 6. Exit ");
            System.out.println("Enter your choice :");
            ch = sc.nextInt();
            switch(ch) {
                case 1:
                    for( int i=0; i< c.length ; i++) {
                        c[i].showAccount();
                    }
```

```
        break;
    case 2:
        System.out.println("Enter account no. you want to
                                search :");
        String ac_no = sc.next();
        boolean found = false;
        for(int i=0; i< c.length ; i++){
            found = c[i].search(ac_no);
            if(found){
                break;
            }
        }
        if(!found){
            System.out.println("Search failed! Account
                                doesn't exist..!!");
        }
        break;

    case 4:
        System.out.println("Enter account no. :");
        ac_no = sc.next(); boolean found;
        found = false;
        for(int i=0; i< c.length ; i++){
            found = c[i].search(ac_no);
            if(found){
                c[i].deposit();
                break;
            }
```

```
        }
        if (! found) {
            System.out.println("Search failed! Account
                                doesn't exist...!!");
        }
        break;
    case 5:
        System.out.print("Enter account no:");
        ac_no = sc.next();
        boolean found = false;
        for(int i=0; i< c.length; i++) {
            found = c[i].search(ac_no);
            if(found) {
                c[i].withdrawal();
                break;
            }
        }
        if(! found) {
            System.out.println("Search failed! Account
                                doesn't exist...!!");
        }
        break;
    case 6:
        System.out.println("See you soon...");
        break;
    }
  } while(ch!=6);
}}
```

Output 1:

How many number of customers do you want ? 2

Enter Account No : 111

Enter Account type : Savings

Enter Name : Swapnil

Enter Balance : 56900

Enter Account No : 121

Enter Account type : current

Enter Name : Vaibhav

Enter Balance : 20000

BANKING APPLICATION

1. Display all account details
2. Search by account number
3. Search by CIF
4. Deposit the amount.
5. Withdraw the amount.
6. Exit

Output - 2

```
Enter your choice :
1
Name of account holder : Swapnil
Account no. : 111
Account type : Savings
Balance : 56900
Name of account holder : Vaibhav
Account no. : 121
Account type : current
Balance : 20000

Enter your choice :
2
Enter account no. you want to search : 111
Name of account holder : Swapnil
Account no : 111
Account type : Savings
Balance : 56900
```

3.) Write a program in java showing the action from three threads using a suitable example.

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

        Thread td1 = new Thread (new myClass1 ());
        Thread td2 = new Thread (new myClass2());
        Thread td3 = new Thread (new myClass3());
        td1.start();
        td2.start();
        td3.start();
    }
}
class myClass1 implements Runnable {
    public void run () {
        int a = 9 , b = 10;
        System.out.println (a+b);
    }
}
class myClass2 implements Runnable {
    public void run () {
        int a = 9 , b = 10;
        System.out.println (a*b);
    }
}
```

```
class myClass3 implements Runnable {
    public void run() {
        int a = 9, b = 10;
        System.out.println(a-b);
    }
}
```

Output:

```
19
-1
90
```

4). Write a program in java which converts a text file into all capital letters.

```
import java.io.*;
public class Main {
    public static void main (String args[]) throws
                                        IOException
    {
        File file1 = new File("input.txt");
        File file2 = new File("output.txt");
        BufferedReader in = new BufferedReader (
                            new FileReader (file1));
        PrintWriter out = new PrintWriter (new
                                FileWriter (file2));
        int ch;
        while ((ch = in.read()) != -1)
        {
            if (Character.isLowerCase(ch))
            {
                ch = Character.toUpperCase(ch);
            }
            out.write (ch);
        }
        in.close();
        out.close();
    }
}
```

input . txt :

Hello World

output . txt :

HELLO WORLD .

5). Create a person class with private instance variable for Person's name and birth date. Add appropriate accessor methods to access the variables. Then create a subclass CollegeGraduate with private instance variables for the student's GPA and year of graduation and appropriate accessors for these variables. Don't forget to include appropriate constructors for your classes. Then create a class with main method that manages your classes.

```
public class Main {
    public static void main (String args[]) {
        CollegeGraduate cg = new CollegeGraduate
                        (9, "2023", "Swapnil", "10-03-2001");
        System.out.println (cg.getName() + " " +
                            cg.getDOB() + " " +
                            cg.getGPA() + " " +
                            cg.getYearOfCompletion());

        Person p = new Person ("Vaibhav", "07-06-2006");
        System.out.println (p.getName() + " " +
                            p.getDOB());
    }
}
class Person {
    private String name;
    private String DOB;
```

```
    public Person ( String name, String DOB) {
            this . name = name;
            this . DOB = DOB ;
    }
    public String getName () {
            return name;
    }
     public String getDOB () {
            return DOB;
     }
}
class CollegeGraduate extends Person {
    private float GPA;
    private String yearOfCompletion;
    public CollegeGraduate ( float GPA, String yearOfCompletion
                            , String name, String DOB)
     {
          super ( name, DOB);
          this . GPA = GPA;
          this . yearOfCompletion = yearOfCompletion;
     }
     public float getGPA() {
            return GPA;
     }
     public String getYearOfCompletion() {
            return yearOfCompletion;
     }
}
```

Output:

Swapnil 10-03-2001 9.0 2023

Vaibhav 07-06-2006

6). Write an applet which draws a human face with ovals and arcs.

```
import java.awt.*;
import java.applet.*;
public class Human_Face extends Applet
{
    public void paint (Graphics g)
    {
        g.drawOval (40, 40, 120, 150);   // Head
        g.drawOval (57, 75, 30, 20);     // Left Eye
        g.drawOval (110, 75, 30, 20);    // Right Eye
        g.fillOval (68, 81, 10, 10);     // pupil (left)
        g.fillOval (121, 81, 10, 10);    // pupil (right)
        g.drawOval (85, 100, 30, 30);    // Nose
        g.fillArc (60, 125, 80, 40, 180, 180)  // Mouth
        g.drawOval (25, 92, 15, 30);     // Left Ear
        g.drawOval (160, 92, 15, 30);    // Right Ear
    }
}
```

```
<html>
 <head> </head>
 <body>
    <applet code = "Human_Face.class" width = "320"
            height = "480"> </applet>
 </body>
</html>
```

7). Write a program in java to show the mouse click event. The program should change the background color of window randomly at each mouse click.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
public class Change_Background extends Applet implements
                                                KeyListener
{
    public void init()
    {
        setBackground (Color.white);
        addKeyListener (this);
    }
    public void keyPressed (KeyEvent e)
    {
        if (e.getKeyCode() == KeyEvent.VK_CONTROL)
        {
            int R = (int) (Math.random() * 100) % 255;
            int G = (int) (Math.random() * 100) % 255;
            int B = (int) (Math.random() * 100) % 255;
            Color mycolor = new Color(R, G, B);
            this.setBackground (mycolor);
        }
    }
```

```
    // Empty function
    public void keyReleased (KeyEvent a)
    {
    }

    // Empty function
    public void keyTyped (KeyEvent a)
    {
    }
}
```