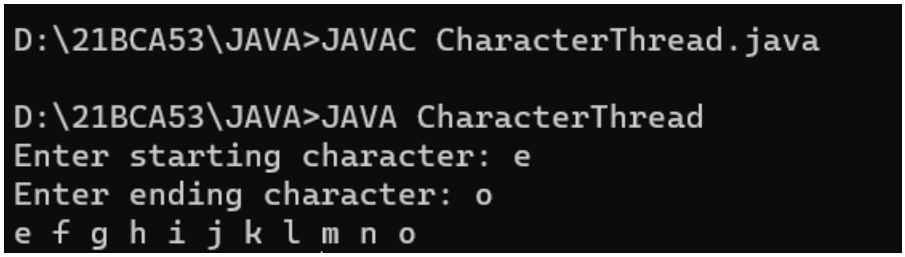
1. **Write a program which accepts starting character and ending character. Display one by one character from starting character till the ending character at the interval of one second using thread.**

CODING

```
import java.util.Scanner;

public class CharacterThread implements Runnable {
    private char startChar;
    private char endChar;

    public CharacterThread(char startChar, char endChar) {
        this.startChar = startChar;
        this.endChar = endChar;
    }

    @Override
    public void run() {
        try {
            for (char c = startChar; c <= endChar; c++) {
                System.out.print(c + " ");
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

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

        System.out.print("Enter starting character: ");
        char startChar = scanner.next().charAt(0);
```

```
        System.out.print("Enter ending character: ");
        char endChar = scanner.next().charAt(0);

        CharacterThread characterThread = new CharacterThread(startChar,
endChar);
        Thread thread = new Thread(characterThread);
        thread.start();
    }
}
```

OUTPUT



**2. Write a program that stores details of 5 employees and display this information after every 10 second.**

CODING

```
import java.util.ArrayList;

public class EmployeeDetails {

    public static void main(String[] args) {

        // Create an ArrayList to store employee objects
        ArrayList<Employee> employees = new ArrayList<Employee>();

        // Add 5 employees to the ArrayList
        employees.add(new Employee("John", "Doe", "john.doe@example.com"));
        employees.add(new Employee("Jane", "Doe", "jane.doe@example.com"));
        employees.add(new Employee("Bob", "Smith", "bob.smith@example.com"));
        employees.add(new Employee("Alice", "Johnson",
"alice.johnson@example.com"));
        employees.add(new Employee("Tom", "Brown",
"tom.brown@example.com"));

        // Create a new thread to display employee information every 10 seconds
        Thread displayThread = new Thread(new Runnable() {
            public void run() {
                while (true) {
                    System.out.println("Employee Details:");
                    for (Employee emp : employees) {
                        System.out.println(emp);
                    }
                    System.out.println();
                    try {
                        Thread.sleep(10000); // Wait for 10 seconds
                    } catch (InterruptedException e) {
                        e.printStackTrace();
```

```
                }
            }
        }
    });

    // Start the thread
    displayThread.start();
  }
}

// Employee class with fields for first name, last name, and email
class Employee {
  String firstName;
  String lastName;
  String email;

  public Employee(String firstName, String lastName, String email) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.email = email;
  }

  public String toString() {
    return firstName + " " + lastName + " (" + email + ")";
  }
}
```

OUTPUT

```
D:\21BCA53\JAVA>javac EmployeeDetails.java

D:\21BCA53\JAVA>java EmployeeDetails
Employee Details:
John Doe (john.doe@example.com)
Jane Doe (jane.doe@example.com)
Bob Smith (bob.smith@example.com)
Alice Johnson (alice.johnson@example.com)
Tom Brown (tom.brown@example.com)
```

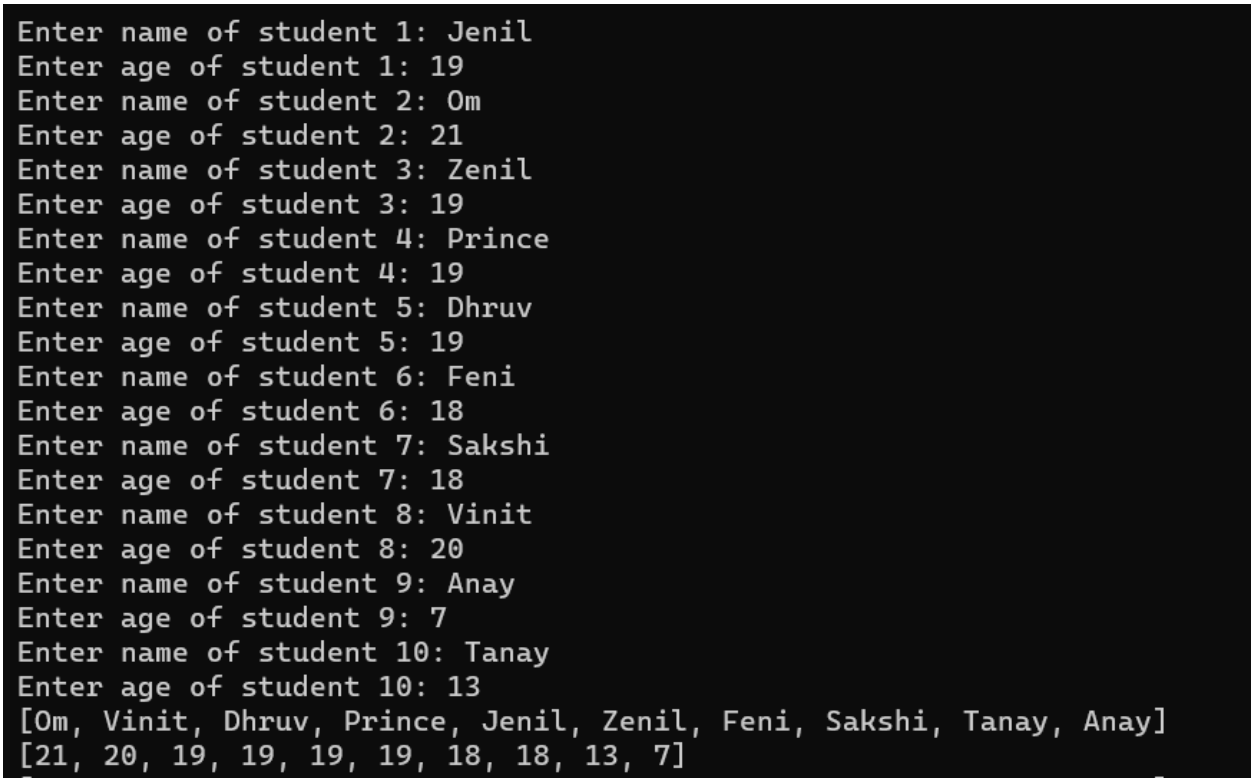
**3. Write a java application which accepts 10 names of student and their age. Sort names and age in descending order at an interval of 1 second using thread.**

CODING

```
import java.util.*;

public class StudentSorter implements Runnable {
   private String[] names;
   private int[] ages;

   public StudentSorter(String[] names, int[] ages) {
      this.names = names;
      this.ages = ages;
   }

   @Override
   public void run() {
      for (int i = 0; i < 10; i++) {
         try {
            Thread.sleep(1000); // wait for 1 second
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
         sortDescending(); // sort the names and ages in descending order
         System.out.println(Arrays.toString(names));
         System.out.println(Arrays.toString(ages));
      }
   }

   private void sortDescending() {
      Map<String, Integer> map = new HashMap<>();
      for (int i = 0; i < 10; i++) {
         map.put(names[i], ages[i]); // create a map of names and ages
      }
```

```
        List<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());
        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String,
Integer> o2) {
                return (o2.getValue()).compareTo(o1.getValue()); // sort in descending
order based on age
            }
        });
        for (int i = 0; i < 10; i++) {
            Map.Entry<String, Integer> entry = list.get(i);
            names[i] = entry.getKey(); // update the sorted names array
            ages[i] = entry.getValue(); // update the sorted ages array
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String[] names = new String[10];
        int[] ages = new int[10];
        for (int i = 0; i < 10; i++) {
            System.out.print("Enter name of student " + (i+1) + ": ");
            names[i] = scanner.next();
            System.out.print("Enter age of student " + (i+1) + ": ");
            ages[i] = scanner.nextInt();
        }
        StudentSorter sorter = new StudentSorter(names, ages);
        Thread thread = new Thread(sorter);
        thread.start();
    }
}
```

OUTPUT




**4. Create package stores. Under it create a class called stock with member variable (item_no, item_name, stock_availible, and cost). Under the default package create a class called sales with field name (qty_sold) and it is the child class of stores class. Write a program to print the current stock of each item and perform addition.**

**5. Create a class namely Vehicle to maintain vehicle data like chassisNo, nameOfVehicle, colour, owner using singly circular linked list. Perform following operations on student list: a. Add vehicle details at the end of the list b. Remove last vehicle detail in the list c. Display all vehicle details in the list**

CODING

```
import java.lang.*;
 class Vehicle {
    private int chassisNo;
```

```
    private String nameOfVehicle;
    private String colour;
    private String owner;
    private Vehicle next;

    public Vehicle(int chassisNo, String nameOfVehicle, String colour, String
owner) {
        this.chassisNo = chassisNo;
        this.nameOfVehicle = nameOfVehicle;
        this.colour = colour;
        this.owner = owner;
        this.next = null;
    }

    public void setNext(Vehicle next) {
        this.next = next;
    }

    public Vehicle getNext() {
        return next;
    }

    public void addVehicle(Vehicle vehicle) {
        Vehicle current = this;
        while (current.next == this) {
            current = current.next;
        }
        current.next = vehicle;
        vehicle.next = this;
    }

    public void removeLastVehicle() {
        Vehicle current = this;
        while (current.next.next != this) {
            current = current.next;
```

```
        }
        current.next = this;
    }

    public void displayAllVehicles() {
        Vehicle current = this;
        do {
            System.out.println("Chassis No: " + current.chassisNo);
            System.out.println("Name of Vehicle: " + current.nameOfVehicle);
            System.out.println("Colour: " + current.colour);
            System.out.println("Owner: " + current.owner);
            System.out.println("------------------------");
            current = current.next;
        } while (current != this);
    }
}
//MAIN METHOD:
public class vehicle_detail {
    public static void main(String[] args) {
        Vehicle vehicle1 = new Vehicle(1234, "Car", "Red", "John");
        Vehicle vehicle2 = new Vehicle(5678, "Motorcycle", "Blue", "Jane");
        Vehicle vehicle3 = new Vehicle(9012, "Truck", "Green", "Bob");

        vehicle1.addVehicle(vehicle2);
        vehicle1.addVehicle(vehicle3);

        vehicle1.displayAllVehicles();

        vehicle1.removeLastVehicle();

        vehicle1.displayAllVehicles();
    }
}
```

OUTPUT

```
D:\21BCA53\JAVA>javac vehicle_detail.java

D:\21BCA53\JAVA>java vehicle_detail
Chassis No: 1234
Name of Vehicle: Car
Colour: Red
Owner: John
-------------------------
Chassis No: 9012
Name of Vehicle: Truck
Colour: Green
Owner: Bob
-------------------------
Chassis No: 1234
Name of Vehicle: Car
Colour: Red
Owner: John
-------------------------
```

**6. Create a class namely Book to maintain Book details like id, name, quantity and author using singly linked list. Perform following operations on book list: a. Add book details in the begging of the list b. Add book details at the end of the list c. Add book detail at particular position d. Remove first book detail from the list e. Remove last book detail from the list f. Display all book details in the list**

CODING

**7. Write a programme to draw smiley with colour using applet.**

CODING

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class Smiley extends Applet {
        /*<applet code="Smiley.class" width="100" height="100"></applet>*/

    public void paint(Graphics g) {
        // draw face
        g.setColor(Color.YELLOW);
        g.fillOval(50, 50, 200, 200);
        g.setColor(Color.BLACK);
        g.drawOval(50, 50, 200, 200);

        // draw eyes
        g.setColor(Color.BLUE);
        g.fillOval(100, 100, 25, 25);
        g.fillOval(175, 100, 25, 25);

        // draw mouth
        g.setColor(Color.RED);
        g.fillArc(75, 150, 150, 50, 180, 180);
    }
}
```

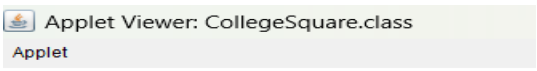
OUTPUT

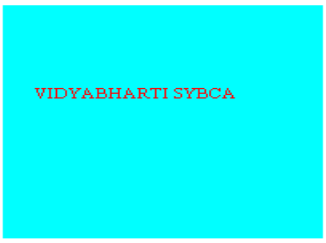
**8. Create an applet which displays a solid square having red colour. Display name of your college within the square with font style =’Times New Roman’, font size=12 and font colour=’Yellow’.**

CODING

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;

public class CollegeSquare extends Applet {
        /*<applet code="CollegeSquare.class" width="100"
height="100"></applet>*/

    public void paint(Graphics g) {
        // draw square
        g.setColor(Color.cyan);
        g.fillRect(50, 50, 200, 200);
        // set font
        Font font = new Font("Times New Roman", Font.PLAIN, 12);
        g.setFont(font);
        // draw text
        g.setColor(Color.red);
        g.drawString("VIDYABHARI SYBCA", 70, 130);
    }
}
```

OUTPUT



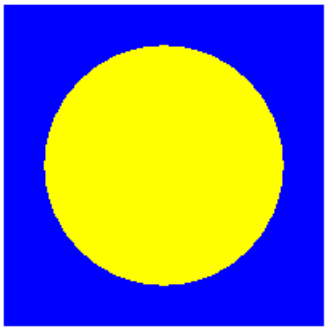



**9. Write a program to draw circle inside a square in applet with different colours**

CODING

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class CircleSquare extends Applet {
/*<applet code="CircleSquare.class" width="100" height="100"></applet>*/

    public void paint(Graphics g) {
        // draw square
        g.setColor(Color.BLUE);
        g.fillRect(50, 50, 200, 200);

        // draw circle
        g.setColor(Color.YELLOW);
        g.fillOval(75, 75, 150, 150);
    }
}
```

OUTPUT



**10. Write an applet program which accepts number of ovals user wants to display using parameter tag and draws ovals in different positions.**

CODING

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.util.Random;

public class OvalApplet extends Applet {
  /*  <applet code="OvalApplet.class" width="500" height="500">
        <param name="numOvals" value="5">
     </applet>*/
   int numOvals;

   public void init() {
     String num = getParameter("numOvals");
     numOvals = Integer.parseInt(num);
   }

   public void paint(Graphics g) {
     for(int i = 1; i <= numOvals; i++) {
       int x = (int)(Math.random() * getWidth());
       int y = (int)(Math.random() * getHeight());
       int width = (int)(Math.random() * 100);
       int height = (int)(Math.random() * 100);
       g.drawOval(x, y, width, height);
     }
   }

}
```

OUTPUT