

### Program-1

**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.**

Code:

```
import java.util.*;
public class CharacterPrinter implements Runnable {
    private char startChar;
    private char endChar;

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

    public void run() {
        for (char ch = startChar; ch <= endChar; ch++) {
            System.out.print(ch + " ");
            try {
                Thread.sleep(1000); // Wait for one second
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    public static void main(String[] args) {
        Scanner s=new Scanner (System.in);
        System.out.println("Enter the starting Character:");
        char startChar =s.next().charAt(0);
        System.out.println("Enter the Endding Character:");
        char endChar = s.next().charAt(0);

        CharacterPrinter characterPrinter = new CharacterPrinter(startChar, endChar);
        Thread thread = new Thread(characterPrinter);
        thread.start();
    }
}
```

Output:

```
D:\java\21BCA73>javac CharacterPrinter.java
D:\java\21BCA73>java CharacterPrinter Enter the starting Character:
A
Enter the Ending Character:
K
A B C D E F G H I J K
D:\java\21BCA73>
```

### Program-2

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

Code:

```
import java.util.*;
public class Empdetail implements Runnable {
    int id[]={ 1,2,3,4,5};
    String name[]={ "Kaushik", "Abhay", "Ajay", "Chandresh", "Brijesh"};
    String dept[]={ "IT", "Management", "Traansport", "HR", "Salse"};
    int salary[]={ 50000,45000,40000,52000,48000};
    public void run() {

        for (int i=0;i<5;i++) {
            System.out.println("Emp id:"+id[i]);
            System.out.println("Emp name:"+name[i]);
            System.out.println("Emp department:"+dept[i]);
            System.out.println("Emp salary:"+salary[i]);
            try {
                Thread.sleep(1000); // Wait for one second
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

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

```
Empdetail Empdetail = new Empdetail();
Thread thread = new Thread(Empdetail);
thread.start();
}
}
```

Output:

```
D:\java\21BCA73>javac Empdetail.java
D:\java\21BCA73>java Empdetail
Emp id:1
Emp name: Kaushik
Emp department: IT
Emp salary: 50000 Emp id:2
Emp name: Abhay
Emp department: Management
Emp salary: 45000
Emp id:3
Emp name: Ajay
Emp department: Traansport Emp salary: 40000 Emp id:4
Emp name: Chandresh
Emp department: HR Emp salary: 52000 Emp id:5
Emp name: Brijesh Emp department: Salse Emp salary: 48000
D:\java\21BCA73>
```

### Program-3

1. 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.

Code:

```
import java.util.Arrays;
import java.util.Scanner;

public class Studentdetail1 {

    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.nextLine();
System.out.print("Enter age of student " + (i + 1) + ": ");
ages[i] = scanner.nextInt();
scanner.nextLine();
}

while (true) {
    System.out.println("\nSelect an option:");
    System.out.println("1. Sort via Name.");
    System.out.println("2. Sort via Age.");
    System.out.println("3. Exit");
    System.out.print("\nSelect Your Choice : ");
    int choice = scanner.nextInt();
    scanner.nextLine();

    switch (choice) {
        case 1:
            for (int i = 0; i < 10; i++) {
                for (int j = i + 1; j < 10; j++) {
                    if
(names[i].compareToIgnoreCase(names[j]) < 0) {
                        String tempName = names[i];
                        names[i] = names[j];
                        names[j] = tempName;
                        int tempAge = ages[i];
                        ages[i] = ages[j];
```

```
        ages[j] = tempAge;
    }
}

System.out.println("\nSorted Names in Descending Order:");
for (int i = 0; i < 10; i++) {
    try {
        System.out.println(names[i] + " - " + ages[i]);
        Thread.sleep(1000);
    }
    catch (InterruptedException e) {
        e.printStackTrace();
    }
}
break;
case 2:
    for (int i = 0; i < 10; i++) {
        for (int j = i + 1; j < 10; j++) {
            if (ages[i] < (ages[j])) {
                int tempage = ages[i];
                ages[i] = ages[j];
                ages[j] = tempage;
                String tempname = names[i];
                names[i] = names[j];
                names[j] = tempname;
            }
        }
    }
    System.out.println("\nSorted Ages in Descending Order:");
    for (int i = 0; i < 10; i++) {
        try {
            System.out.println(ages[i] + " - " + names[i]);
            Thread.sleep(1000);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    break;
    case 3:
        System.out.println("Exiting program...");
        System.exit(0);
        break;
    default:
        System.out.println("Invalid choice. Try again.");
}
```

```

    }
}
}

```

Output:

```

D:\java\21BCA73>javac Studentdetail.java
D:\java\21BCA73>java Studentdetail1
Enter name of student 1: Yash
Enter age of student 1: 18 Enter name of student 2: Vivek
Enter age of student 2: 19
Enter name of student 3: Chandresh Enter age of student 3: 20
Enter name of student 4: Kaushik Enter age of student 4: 19 Enter name of student 5: Chetan Enter age of
student 5: 19 Enter name of student 6: Brijesh Enter age of student 6: 18 Enter name of student 7:
Shivraj Enter age of student 7: 18 Enter name of student 8: Abhay Enter age of student 8: 19 Enter name
of student 9: Ajay Enter age of student 9: 19
Enter name of student 10: Kamlesh Enter age of student 10: 21
Select an option:
1. Sort via Name.
2. Sort via Age.
3. Exit
Select Your Choice: 1
Sorted Names in Descending Order: Yash - 18
Vivek
—
19
Shivraj - 18
Kaushik
—

```

```

Select an option: 1. Sort via Name. 2. Sort via Age. 3. Exit
Select Your Choice : 2
Sorted Ages in Descending Order: 21 - Kamlesh
—
20 Chandresh
19 - Vivek
19 - Chetan 19 - Kaushik
19 - Ajay
19 - Abhay 18 - Brijesh 18 - Yash 18 - Shivraj
Select an option: 1. Sort via Name. 2. Sort via Age. 3. Exit
Select Your Choice :

```

### Program -4

**3. Create package stores. Under it create a class called stock With member variable (item\_no, item\_name, stock\_available, 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.**

Code:

```
package stores;
```

```

public class stock {
    private int item_no;
    private String item_name;

```

```
private int stock_available;
private double cost;

public stock(int item_no, String item_name, int stock_available, double cost) {
    this.item_no = item_no;
    this.item_name = item_name;
    this.stock_available = stock_available;
    this.cost = cost;
}

public int getItem_no() {
    return item_no;
}

public String getItem_name() {
    return item_name;
}

public int getStock_available() {
    return stock_available;
}

public double getCost() {
    return cost;
}

public void setStock_available(int stock_available) {
    this.stock_available = stock_available;
}

public void setCost(double cost) {
    this.cost = cost;
}

public String toString() {
    return "Item No.: " + item_no + ", Item Name: " + item_name + ", Stock Available: " +
stock_available + ", Cost: " + cost;
}
}
package stores;
public class sales extends stores.stock {
    private int qty_sold;

    public sales(int item_no, String item_name, int stock_available, double cost, int qty_sold)
    {
        super(item_no, item_name, stock_available, cost);
    }
}
```

```
        this.qty_sold = qty_sold;
    }

    public int getQty_sold() {
        return qty_sold;
    }

    public void setQty_sold(int qty_sold) {
        this.qty_sold = qty_sold;
    }

    public double calculateSales() {
        return qty_sold * getCost();
    }

    public String toString() {
        return super.toString() + ", Quantity Sold: " + qty_sold + ", Sales: " + calculateSales();
    }
}

import stores.*;
import java.util.*;

public class Program4{
    public static void main(String[] args) {
        ArrayList<stock> items = new ArrayList<stock>();
        items.add(new stock(1, "Mango", 10, 20.0));
        items.add(new stock(2, "Apple", 20, 30.0));
        items.add(new stock(3, "Banana", 30, 40.0));

        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nCurrent Stock:");
            for (stock item : items) {
                System.out.println(item);
            }

            System.out.print("\nEnter the item no. to add stock, or 0 to exit:");
            int item_no = scanner.nextInt();
            if (item_no == 0) {
                break;
            }

            stock item = items.stream().filter(i -> i.getItem_no() ==
item_no).findFirst().orElse(null);
            if (item == null) {
```



```

        System.out.println("Invalid item no.");
    } else {
        System.out.print("\nEnter the quantity to add:");
        int qty = scanner.nextInt();
        item.setStock_available(item.getStock_available() + qty);
        System.out.println("Stock added successfully.");
    }
}
}
}
}

```

Output:

```

D:\java\21BCA73>javac Program4.java
D:\java\21BCA73>java Program4
Current Stock:
Item No.: 1, Item Name: Mango, Stock Available: 10, Cost: 20.0 Item No.: 2, Item Name: Apple, Stock
Available: 20, Cost: 30.0 Item No. 3, Item Name: Banana, Stock Available: 30, Cost: 40.0
Enter the item no. to add stock, or to exit:3
Enter the quantity to add:5
Stock added successfully.
Current Stock:
Item No.: 1, Item Name: Mango, Stock Available: 10, Cost: 20.0 Item No.: 2, Item Name: Apple, Stock
Available: 20, Cost: 30.0 Item No.: 3, Item Name: Banana, Stock Available: 35, Cost: 40.0
Enter the item no. to add stock, or to exit:0
D:\java\21BCA73>

```

### Program-5

**4. 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.

Code:

```

import java.util.Scanner;

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;
```

```
        System.out.println("\nData Inserted Successfully.");
```

```
}
```

```
public int getChassisNo() {
```

```
    return chassisNo;
```

```
}
```

```
public void setChassisNo(int chassisNo) {
```

```
        this.chassisNo = chassisNo;
    }

    public String getNameOfVehicle() {
        return nameOfVehicle;
    }

    public void setNameOfVehicle(String nameOfVehicle) {
        this.nameOfVehicle = nameOfVehicle;
    }

    public String getColour() {
        return colour;
    }

    public void setColour(String colour) {
        this.colour = colour;
    }

    public String getOwner() {
        return owner;
    }

    public void setOwner(String owner) {
        this.owner = owner;
    }

    public Vehicle getNext() {
        return next;
    }
```

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

}

class VehicleList {
    private Vehicle tail;

    public VehicleList() {
        tail = null;
    }

    public void addVehicle(int chassisNo, String nameOfVehicle, String colour, String owner)
    {
        Vehicle newVehicle = new Vehicle(chassisNo, nameOfVehicle, colour, owner);
        if (tail == null) {
            tail = newVehicle;
            tail.setNext(tail);
        }
        else {
            newVehicle.setNext(tail.getNext());
            tail.setNext(newVehicle);
            tail = newVehicle;
        }
    }

    public void removeLastVehicle() {
        if (tail == null) {
            System.out.println("List is empty");
        }
    }
}
```

```
        return;
    }

    if (tail.getNext() == tail) {
        tail = null;
        return;
    }

    Vehicle current = tail.getNext();
    while (current.getNext() != tail) {
        current = current.getNext();
    }
    current.setNext(tail.getNext());
    tail = current;
}

public void displayVehicles() {
    if (tail == null) {
        System.out.println("List is empty");
        return;
    }

    Vehicle current = tail.getNext();
    do {
        System.out.println(".....");
        System.out.println("Chassis No: " + current.getChassisNo() +
            "\nName of Vehicle: " + current.getNameOfVehicle() +
            "\nColour: " + current.getColour() +
            "\nOwner: " + current.getOwner());
        System.out.println(".....");
    } while (current.getNext() != null);
}
```

```
        current = current.getNext();
    } while (current != tail.getNext());
}
}
```

```
public class Program5{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        VehicleList vehicleList = new VehicleList();
        while (true) {
            System.out.println("\n.....");
            System.out.println("\nCircular Singly Linked List Operations\n");
            System.out.println(".....");
            System.out.println("1. Insert at End.");
            System.out.println("2. Delete from End.");
            System.out.println("3. Get Item detail's.");
            System.out.println("4. Exit.");
            System.out.println(".....");
            System.out.print("Enter your Choice : ");
            int choice = scan.nextInt();
            switch (choice)
            {
                case 1 :
                    int ch_no;
                    String nameOfVeh, colour, owner;
                    System.out.print("Enter Chassis_No : ");
                    ch_no=scan.nextInt();
                    scan.nextLine();
```

```
        System.out.print("Enter Name of vehicle : ");
        nameOfVeh=scan.nextLine();
        System.out.print("Enter Color of vehicle : ");
        colour=scan.nextLine();
        System.out.print("Enter Owner Name : ");
        owner=scan.nextLine();

        vehicleList.addVehicle(ch_no,nameOfVeh,colour,owner);

        break;
    case 2 :

        vehicleList.removeLastVehicle();

        System.out.println("\nData Deleted Successfully.");

        break;
    case 3 :

        System.out.println("Vehicle details:");
        vehicleList.displayVehicles();

        break;
    case 4 :

        System.out.println("Program Exited...");
        System.exit(0);

        break;
```

default:

```
System.out.println("Invalid choice. Try again.");
```

```
    }
}
}
}
```

Output:

```
D:\java\21BCA73>javac Program5.java
D:\java\21BCA73>java Program5
Circular Singly Linked List Operations
1. Insert at End.
2. Delete from End.
3. Get Item detail's.
4. Exit.
Enter your Choice : 1
Enter Chassis_No : 5
Enter Name of vehicle : Maruti
Enter Color of vehicle: Red
Enter Owner Name: Maruti Suzuki
Data Inserted Successfully.
Circular Singly Linked List Operations
1. Insert at End.
2. Delete from End.
3. Get Item detail's.
4. Exit.
Enter your Choice: 1
```

## Program-6

**5. 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 .

Code:

```
import java.util.Scanner;
class Book {
    int id;
    String name;
    int quantity;
    String author;
    Book next;
```



```
public Book(int id, String name, int quantity, String author) {
    this.id = id;
    this.name = name;
    this.quantity = quantity;
    this.author = author;
    this.next = null;
        System.out.println("\nData Inserted Successfully.");
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getQuantity() {
```

```
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public Book getNext() {
        return next;
    }

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

class BookList {
    private Book head;

    public BookList() {
        head = null;
    }

    public void addBookAtBeginning(int id, String name, int quantity, String author) {
        Book newBook = new Book(id, name, quantity, author);
        newBook.setNext(head);
        head = newBook;
    }

    public void addBookAtEnd(int id, String name, int quantity, String author) {
        Book newBook = new Book(id, name, quantity, author);
        if (head == null) {
            head = newBook;
        } else {
            Book current = head;
            while (current.getNext() != null) {
                current = current.getNext();
            }
        }
    }
}
```

```
    }  
    current.setNext(newBook);  
}  
}
```

```
public void addBookAtPosition(int id, String name, int quantity, String author, int  
position) {
```

```
    if (position == 1) {  
        addBookAtBeginning(id, name, quantity, author);  
    } else {  
        Book newBook = new Book(id, name, quantity, author);  
        Book current = head;  
        int currentPosition = 1;  
        while (currentPosition < position - 1 && current != null) {  
            current = current.getNext();  
            currentPosition++;  
        }  
        if (current != null) {  
            newBook.setNext(current.getNext());  
            current.setNext(newBook);  
        } else {  
            System.out.println("Invalid position");  
        }  
    }  
}
```

```
public void removeFirstBook() {  
    if (head == null) {  
        System.out.println("List is empty");  
    } else {  
        head = head.getNext();  
    }  
}
```

```
public void removeLastBook() {  
    if (head == null) {  
        System.out.println("List is empty");  
    }  
    else if (head.getNext() == null) {  
        head = null;  
    }  
    else {  
        Book current = head;  
        while (current.getNext().getNext() != null) {  
            current = current.getNext();  
        }  
    }  
}
```

```

        current.setNext(null);
    }
}

public void displayBooks() {
    if (head == null) {
        System.out.println("List is empty");
    } else {
        Book current = head;
        System.out.println(".....");
        while (current != null) {
            System.out.println("ID: " + current.getId() + ", Name: " +
current.getName() + ", Quantity: " + current.getQuantity() + ", Author: " +
current.getAuthor());
            current = current.getNext();
        }
        System.out.println(".....");
    }
}
}

```

```

public class Program6
{
    public static void main(String[] args)
    {
        int id;
        String name;
        int quantity;
        String author;
        Scanner scan = new Scanner(System.in);
        BookList bookList = new BookList();
        while (true) {
            System.out.println("\n.....");
            System.out.println("\nSingly Linked List Operations\n");
            System.out.println(".....");
            System.out.println("1. Insert at Beginning.");
            System.out.println("2. Insert at End.");
            System.out.println("3. Insert at Position.");
            System.out.println("4. Delete from Head.");
            System.out.println("5. Delete from Tail.");
            System.out.println("6. Display Data.");
            System.out.println("7. Exit.");
            System.out.println(".....");
            System.out.print("Enter your Choice : ");
            int choice = scan.nextInt();
            switch (choice)

```

```
{
case 1 :

        System.out.print("Enter Your ID : ");
        id=scan.nextInt();
        scan.nextLine();
        System.out.print("Enter Your Name : ");
        name=scan.nextLine();
        System.out.print("Enter Quantity of Books : ");
        quantity=scan.nextInt();
        scan.nextLine();
        System.out.print("Enter Author Name : ");
        author=scan.nextLine();
        bookList.addBookAtBeginning(id,name,quantity,author);

        break;
case 2 :

        System.out.print("Enter Your ID : ");
        id=scan.nextInt();
        scan.nextLine();
        System.out.print("Enter Your Name : ");
        name=scan.nextLine();
        System.out.print("Enter Quantity of Books : ");
        quantity=scan.nextInt();
        scan.nextLine();
        System.out.print("Enter Author Name : ");
        author=scan.nextLine();
        bookList.addBookAtEnd(id,name,quantity,author);

        break;
case 3 :

        int position;
        System.out.print("Enter Position you want to Insert Record : ");
        position=scan.nextInt();
        System.out.print("Enter Your ID : ");
        id=scan.nextInt();
        scan.nextLine();
        System.out.print("Enter Your Name : ");
        name=scan.nextLine();
        System.out.print("Enter Quantity of Books : ");
        quantity=scan.nextInt();
        scan.nextLine();
        System.out.print("Enter Author Name : ");
        author=scan.nextLine();
        bookList.addBookAtPosition(id,name,quantity,author,position);
        break;
case 4 :
        bookList.removeFirstBook();
        System.out.println("\nData Deleted Successfully.");
```

```

        break;
    case 5 :
        bookList.removeLastBook();
        System.out.println("\nData Deleted Successfully.");
        break;
    case 6 :
        bookList.displayBooks();
        break;
        case 7 :
            System.out.println("Program Exited...");
            System.exit(0);
            break;
    default :
        System.out.println("Invalid choice. Try again.");
        break;
    }
}
}

```

Output:

```

D:\java\21BCA73>javac Program6.java
D:\java\21BCA73>java Program6
Singly Linked List Operations
1. Insert at Beginning. 2. Insert at End.
3. Insert at Position. 4. Delete from Head.
5. Delete from Tail. 6. Display Data. 7. Exit.
Enter your Choice: 1
Enter Your ID: 54 Enter Your Name: Yash
Enter Quantity of Books : 5 Enter Author Name : Vtcba
Data Inserted Successfully.
Singly Linked List Operations
1. Insert at Beginning. 2. Insert at End.
3. Insert at Position.
4. Delete from Head.
5. Delete from Tail.
6. Display Data.
7. Exit.
Enter your Choice: 3
Enter Position you want to Insert Record : 0 Enter Your ID: 49

```

## Program-7

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

Code:

```

import java.awt.*;
import java.applet.*;

```

```
/*<applet code="Program7.class" height="800" width="1860"> </applet>*/
```

```
public class Program7 extends Applet {
```

```
    public void paint(Graphics g) {
```

```
        g.setColor(Color.yellow);
```

```
        g.fillOval(50,50,200,200);
```

```
        g.setColor(Color.black);
```

```
        g.drawOval(50,50,200,200);
```

```
        g.setColor(Color.black);
```

```
        g.fillOval(100,100,25,25);
```

```
        g.fillOval(175,100,25,25);
```

```
        g.setColor(Color.black);
```

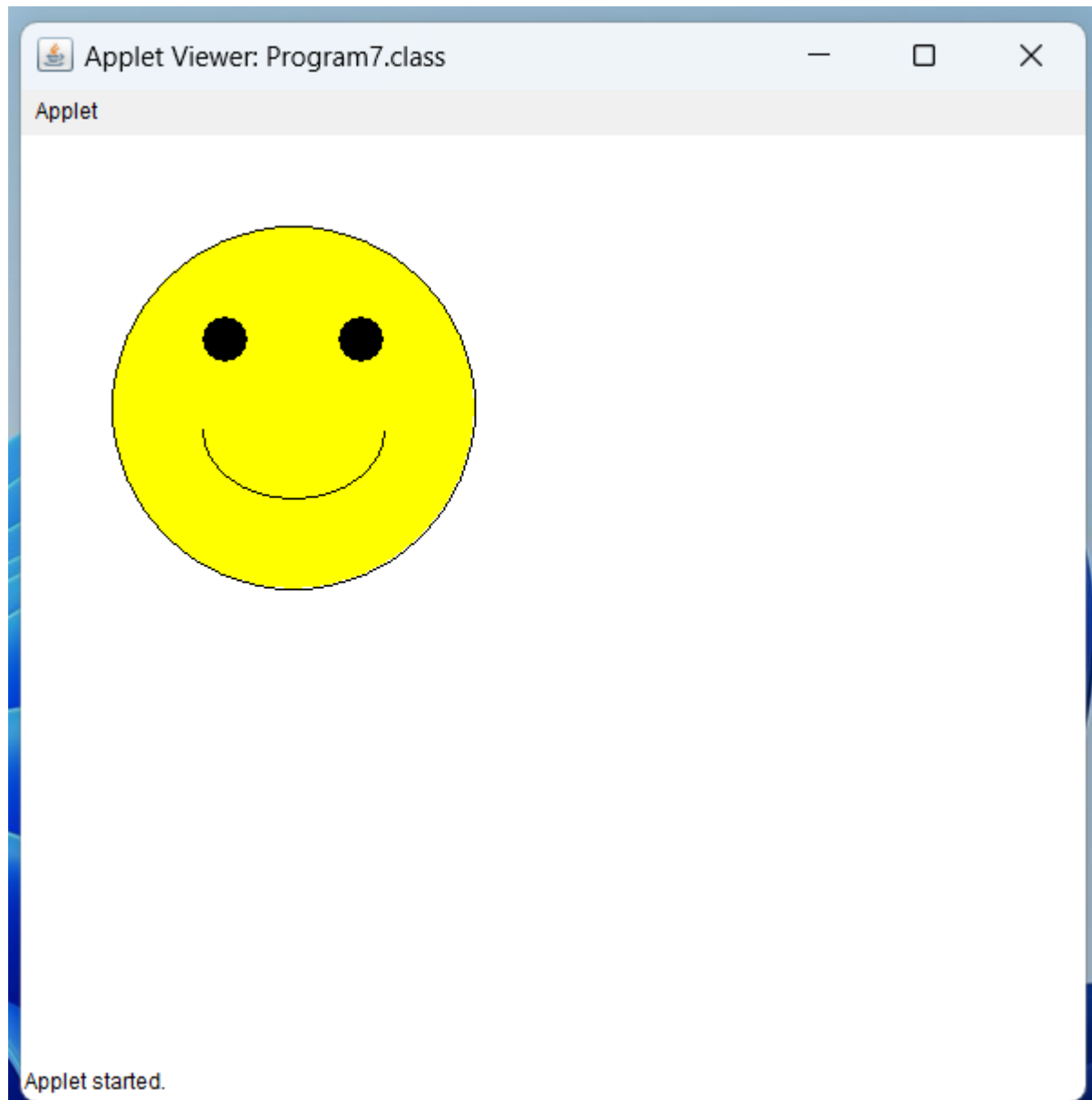
```
        g.drawArc(100,125,100,75,0,-180);
```

```
    }
```

```
}
```

Output:

```
D:\java\21BCA73>javac Program7.java
D:\java\21BCA73>appletviewer Program7.java
```



### Program-8

**7. 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=50 and font Colour='Yellow'.**

Code:

```
import java.awt.*;
```

```
import java.applet.*;
```

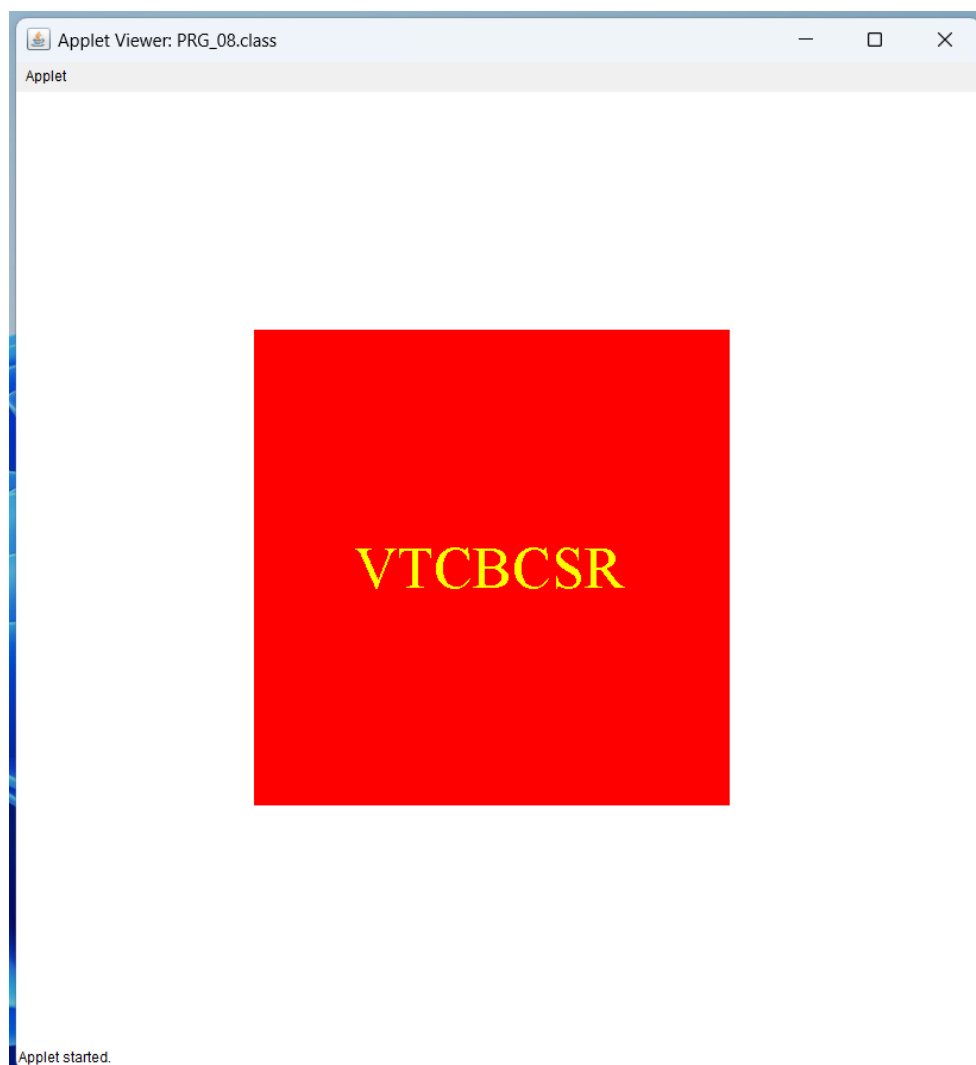
```
//<applet code="Program8.class" height="800" width="1860"> </applet>
```



```
public class Program8 extends Applet {  
  
    public void paint(Graphics g) {  
  
        g.setColor(Color.red);  
        g.fillRect(200,200,400,400);  
  
        g.setColor(Color.yellow);  
        Font font = new Font("Times New Roman", Font.PLAIN, 50);  
        g.setFont(font);  
        FontMetrics metrics = g.getFontMetrics(font);  
        int x = (200 - metrics.stringWidth("My College")) / 2;  
        int y = ((200 - metrics.getHeight()) / 2) + metrics.getAscent();  
        g.drawString("VTCBCSR", 300+x, 300+y);  
  
    }  
}
```

Output:

```
D:\java\21BCA73>javac Program8.java  
D:\java\21BCA73>appletviewer Program8.java
```



**1. Write a program to draw circle inside a square in applet with**

Code:

```
import java.awt.*;
import java.applet.*;

//<applet code="Program9.class" height="800" width="1860"> </applet>

public class Program9 extends Applet {

    public void paint(Graphics g) {

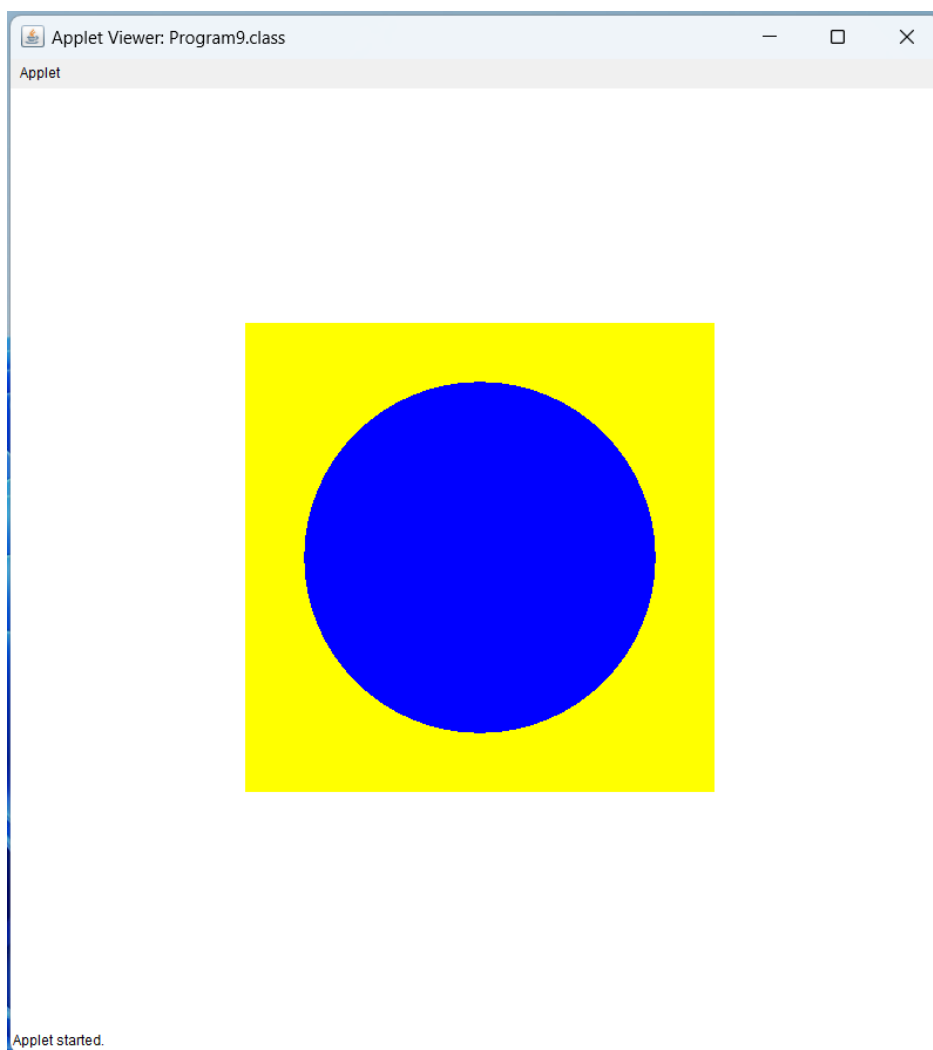
        g.setColor(Color.yellow);
        g.fillRect(200,200,400,400);

        g.setColor(Color.blue);
        g.fillOval(250,250,300,300);

    }
}
```

Output:

```
D:\java\21BCA73>javac Program9.java  
D:\java\21BCA73>appletviewer Program9.java
```



### Program-10

**6. Write an applet program which accepts number of ovals User wants to display using parameter tag and draws ovals in Different positions.**

Code:

```
import java.awt.*;
import java.applet.*;

/*<applet code="Program10.class" height="800" width="1860">
    <param name="numOvals" value="10">
</applet>*/

public class Program10 extends Applet {
    private int numOvals;

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

    public void paint(Graphics g) {
        for (int i = 0; i < numOvals; i++) {
            int x = (int)(Math.random() * 300);
            int y = (int)(Math.random() * 300);
            int w = (int)(Math.random() * 100);
            int h = (int)(Math.random() * 100);
            g.drawOval(x, y, w, h);
        }
    }
}
```

Output:

```
D:\java\21BCA73>javac Program10.java  
D:\java\21BCA73>appletviewer Program10.java
```

