Don Bosco School, Ranchi

Project Work – 2020-21

Class: XII th

Subject: Computer Science

1. Write a menu driven class to accept a number from the user and check whether it is a Palindrome or a Perfect number.

   (i) **Palindrome number** : (A number is a Palindrome which when read in reverse order is same as in the right order)
   Example: 11, 101, 151 etc.

   (ii) **Perfect number** : (A number is called Perfect if it is equal to the sum of its factors other than the number itself.)
   Example: 6 = 1+2+3

   Ans:

```
import java.io.*;
import java.util.*;
class Pali_perfect
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.println("********************MENU*****************");
System.out.println("Press 1:  Check Palindrome Number");
System.out.println("Press 2: Check Perfect Number");
System.out.println("Input Your choice.....");
int ch=sc.nextInt();
switch(ch)
{
case 1:
int d=0,rev=0,n;
System.out.println("Input a number");
n=sc.nextInt();
int n1=n;
while(n>0)
{
d=n%10;
rev=rev*10+d;
n=n/10;
}
if(n1==rev)
System.out.println("Number is Palindrome");
else
System.out.println("Number is not Palindrome");
break;
case 2:
```

```
    int num,s=0;
    System.out.println("input a no.");
    num=sc.nextInt();
    int num1=num;
    for(int i=1;i<=num-1;i++)
    {
    if(num%i==0)
    {
    s=s+i;
    }
    }
    if(s==num1)
    System.out.println("perfect no.");
    else
    System.out.println("not perfect no.");
    break;
default:
    System.out.println("You have chosen wrong choice");

    }
    }
        }
```

2. Write a program in Java to find the sum of the given series:

   i). Sum = $a + \frac{a2}{2} + \frac{a3}{3} + \frac{a4}{4} +$ ……to n terms

   ii). Sum = $\frac{1*2}{1+2} + \frac{2*3}{2+3} + \frac{3*4}{3+4} +$……to n terms

Ans.

```
    import java.io.*;
    import java.util.*;
    class Sum_series
    {
    public static void main(String args[])
    {
    Scanner sc=new Scanner(System.in);
    int a,n;
    double s=0;
    System.out.println("Enter the value of a");
    a=sc.nextInt();
    System.out.println("Enter the value of a");
```

```
n=sc.nextInt();
for(int i=1;i<=n;i++)
{
s=s+Math.pow(a,i)/i;
}
System.out.println("The sum of series="+s");
}
}
```

3. Define a class **Library** with the following description:
   Data Members/Instance variables:
   int acc_num : stores the accession number of the book
   String title : stores the title of the book
   String author : stores the name of the author
   Member methods:
   (i) void input( ) : to input and store the accession number, title and author
   (ii) void compute( ) : to accept the number of days late, calculate and display the fine charged at the rate of Rs. 2 per day.
   (iii) void display( ) : to display the details
   Write a main method to create an object of the class and call the above member methods.
   Ans.

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

public class Library {
    Scanner sc=new Scanner(System.in);
    int acc_num,fine;
    String title,author;
    void input()
    {

        System.out.println("Input accession number");
        acc_num=sc.nextInt();
        System.out.println("Input Title");
        title=sc.next();
        System.out.println("Input Title");
        author=sc.next();

    }
```

```
public void compute()
  {
  System.out.println("Input the number days");
        int d=sc.nextInt();
              fine=2*d;
  }
   public void display()
   {
        System.out.println("Accession number of book="+acc_num);
        System.out.println("Title of the Book="+title);
        System.out.println("Author of the Book="+author);
        System.out.println("Fine="+fine);
   }
   public static void main(String args[])
   {
        Library ob=new Library();
        ob.input();
        ob.compute();
        ob.display();
   }

}
```

4. Define a class named **BookFair** with the following description:
   Instance variables/Data members:
   String Bname : stores the name of the Book
   double price : stores the price of the Book
   Member methods:
   (i) BookFair( ) : default constructor to initialize data members.
   (ii) BookFair(String Bname, double price) : parameterized constructor to initialize data members.
   (iii) void input( ) : to input and store the name and the price of the book.
   (iv) void calculate( ) : to calculate the price after discount. Discount is calculated based on the following criteria.

Price	Discount
Less than or equal to Rs. 1000	2% of price
More than Rs. 1000 and less than or equal to Rs. 3000	10% of price
More than Rs. 3000	15% of price

   (v) void display( ) : to display the name and price of the book after discount.

Write a main method to create an object of the class and call the above member methods.
Ans.

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

```
import java.util.*;

public class BookFair {
	Scanner sc=new Scanner(System.in);
	String bname;
	double price,dis=0;


  public BookFair()
	{
		bname="Julius";
		price=500;
  }
  public BookFair(String bname,double price)
  {
	this.bname=bname;
	this.price=price;
  }
  public void input()
  {
	System.out.println("Input the Book name");
	bname=sc.next();
		System.out.println("Input the Book's price");
	price=sc.nextDouble();

  }
  public void calculate()
  {
  if(price<=1000)
  dis=0.2/100*price;
  else if(price>1000&&price<=3000)
	dis=10.0/100*price;
	else if(price>3000)
		dis=15.0*price;

  }
  public void display()
  {
  System.out.println("Book name="+bname);
  System.out.println("Book price after discount="+dis);

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

```
        BookFair ob=new BookFair();
        ob.calculate();
        ob.display();
        BookFair obj=new BookFair("Hokins",400);
        obj.calculate();
        obj.display();
        BookFair obj1=new BookFair();
        obj1.input();
        obj1.calculate();
        obj1.display();
    }


}
```

5. Design a class to overload a function **series( )** as follows:
   (i) **double series(double n)** with one double argument and returns the sum of the series,

   Sum = $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots\ldots\ldots\ldots\ldots + \frac{1}{n}$

   (ii) **double series(double x, double num)** with two double argument and returns the sum of the series,

   Sum = $\frac{x}{5} + \frac{x}{8} + \frac{x}{11} + \frac{x}{14}\ldots\ldots\ldots\ldots\ldots\frac{x}{num}$terms.

   Ans.

```
import java.io.*;
import java.util.*;
class series_overload
{
double  series(double n)
{
int sum;
for(int i=1;i<=n;i++)
{
sum=sum+1/i;
}
System.out.println("sum of series="+sum);
}
double series(double a,double num)
{
int s;
for(int i=5;i<=num;i=i+5)
{
s=s+x/I;
}
```

```
System.out.println("sum of series="+s);
}
public static void main(String args[])
{
series_overload ob=new series_overload();
ob.series(7);
ob.series(6,7);
}
}
```

6. Using the **switch statement**, write a menu driven program to calculate the maturity amount of a Bank Deposit. The user is given the following options:
   (i) Term Deposit
   (ii) Recurring Deposit
   For option (i) accept principal (P), rate of interest (r), and time period in years (n). Calculate and output the maturity amount (A) receivable using the formula$A = P \left[1+ \frac{r}{100}\right]^n$
   For option (ii) accept Monthly Installment (P), rate of interest (r), and time period in months (n). Calculate and output the maturity amount (A) receivable using the formula
   $$A = P \times n + P \times \frac{n(n+1)}{2} \times \frac{r}{100} \times \frac{1}{12}$$
   For incorrect option, an appropriate error message should be displayed.
   Ans.

```
class BankDeposit
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
double p,r,n,a,a1;
System.out.println("**************MENU*****************");
System.out.println("Option 1.   Term Deposit");
System.out.println("Option 2.   Recurring Deposit");
System.out.println("Input principal,Rate and Time");
p=sc.nextDouble();
r=sc.nextDouble();
n=sc.nextDouble();
System.out.println("Input your choice.....");
int ch=sc.nextInt();
switch(ch)
{
    case 1:a=p*Math.pow((1+r/100),n);
                System.out.println("Maturity amount="+a);
                break;
    case 2:a1=p*n+p*n*(n+1)/2*r/100*1/12;
```

```
                System.out.println("Maturity amount="+a1);
                break;
        default: System.out.println("Invalid choice.....");
                break;
    }
    }
    }
```

7. Define a class called **Mobike** with the following description:
   Instance variables/Data members:
   intbno : to store the bike's number
   intphno : to store the phone number of the customer
   String name : to store the name of the customer
   int days : to store the number of days the bike in taken on rent
   int charge : to calculate and store the rental charge
   Member methods:
   (i) void input( ) : to input and store the detail of the customer.
   (ii) void compute( ) : to compute the rental charge
   The rent for a mobike is charged on the following basis.
   - First five days Rs. 500 per day
   - Next five days Rs. 400 per day
   - Rest of the days Rs. 200 per day

   (iii) void display( ) : to display the details of data members.

Write a main method to create an object of the class and call the above member methods.
Ans.

```
class Mobike
{
        int bno,phno,days,charge;
        String name;
        void input()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Input bike number");
                bno=sc.nextInt();
                        System.out.println("Input phone number");
                phno=sc.nextInt();
                        System.out.println("Input customer name");
                name=sc.next();
                        System.out.println("Input the number of Days");
                days=sc.nextInt();

        }
        void compute()
        {
```

```
            if(days<=5)
                charge=days*500;
                else if(days>5&&days<=10)
                    charge=5*500+400*(days-5);
                    else if(days>10)
                        charge=5*500+400*5+200*(days-10);
    }
    void display()
    {
    System.out.println("Bike number="+bno);
        System.out.println("Phone number="+phno);
        System.out.println("Customer name="+name);
        System.out.println("Days="+days);
        System.out.println("Rental Charge="+days);
    }
    public static void main(String args[])
    {
        Mobike ob=new Mobike();
        ob.input();
        ob.compute();
        ob.display();
    }
}
```

8.A class Matrix contains a two dimensional integer array of order [mxn].The maximum value possible for both 'm' and 'n' is 25.Design a class Matrix to find the difference of the two matrices. The details of the members of the class are given below:

Class name	**Matrix**
Data members/Instance variables	
arr1[][],arr2[]	Stores the matrix element
m	Integer to store the number of rows
n	Integer to store the number of columns
Member functions	
void fillarray()	To enter the elements of the matrix
void submat()	Subtract matrix
void display()	Display the matrix elements after subtracting.

Ans.

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

public class Fill_Matrix
 {
```

```
        Scanner sc=new Scanner(System.in);
                int m=3,n=3;

        int arr1[][]=new int[m][n];
        int arr2[][]=new int[m][n];
        int sub[][]=new int[m][n];
void fillarray()
        {
                System.out.println("Enter first matrix:");
                for(int i=0;i<m;i++)
                {
                        for(int j=0;j<n;j++)
                        {
                                arr1[i][j]=sc.nextInt();
                        }
                }
                        System.out.println("Enter second matrix:");
                for(int i=0;i<m;i++)
                {
                        for(int j=0;j<n;j++)
                        {
                                arr2[i][j]=sc.nextInt();
                        }
                }

        }
        void submat()
        {
                System.out.println("Calculating Sum.....");
                for(int i=0;i<m;i++)
                {
                        for(int j=0;j<n;j++)
                        {
                                sub[i][j]=arr1[i][j]-arr2[i][j];
                        }
                }
        }
        void display()
        {
        System.out.println("Print first Matrix:    ");
        for(int i=0;i<m;i++)
                {
                        for(int j=0;j<n;j++)
                        {
```

```
                        System.out.println(arr1[i][j]+" ");
                    }
                    System.out.println();

            }
            System.out.println();
            System.out.println();
            System.out.println("Print second Matrix:     ");
            for(int i=0;i<m;i++)
                    {
                        for(int j=0;j<n;j++)
                        {
                            System.out.println(arr2[i][j]+" ");
                        }
    System.out.println();

            }System.out.println();
            System.out.println();
    System.out.println("Print third matrix after subtravtion Matrix:     ");
            for(int i=0;i<m;i++)
                    {
                        for(int j=0;j<n;j++)
                        {
                            System.out.println(sub[i][j]+" ");
                        }
    System.out.println();

            }System.out.println();
            System.out.println();
            }
            public static void main(String arg[])
            {
                Fill_Matrix ob=new Fill_Matrix();
                ob.fillarray();
                ob.submat();
                ob.display();
            }
    }
```

9.A transpose of an array is obtained by interchanging the elements of the rows and columns.

Class name	**Transarray**
Datamembers	
arr[][]	Stores the matrix elements
m	Integer to store the number of rows

n	Integer to store the number of columns
Member functions	
void fillarray()	To enter the elements of the matrix
void transpose()	To calculate transpose of a given matrix
void display()	Display the array in matrix form

Ans.

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


public class transposematrix {

   public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                int arr[][]=new int[4][4];
                int arr1[][]=new int[4][4];
                int ld=0,rd=0,k=3;
                System.out.println("Input 4x4 matrix");
                for(int i=0;i<4;i++)
                {
                        for(int j=0;j<4;j++)
                        {
                                arr[i][j]=sc.nextInt();
                        }
                        System.out.println("------");
                }
                System.out.println("Element of 4x4 matrix are");
                for(int i=0;i<4;i++)
                {
                        for(int j=0;j<4;j++)
                        {
                        System.out.print(arr[i][j]+" ")         ;
   }
        System.out.println();


}
System.out.println("Transpose of matrix are.....");
for(int i=0;i<4;i++)
                {
                        for(int j=0;j<4;j++)
                        {
                                arr1[j][i]=arr[i][j];
```

```
                }
            }
            for(int i=0;i<4;i++)
            {
                for(int j=0;j<4;j++)
                {
                System.out.print(arr1[i][j]+" ");
    }
        System.out.println();


    }

}
}
```

10.Design a class Stringfun to perform various operations on string

Class name	Stringfun
Datamembers	
str	To store the string
Member functions	
void input()	To accept the string
void words()	To find and display the number of words, number of vowels, and number of uppercase characters in the string
void reverse()	To check and display the reverse String

Ans.

```
import java.io.*;
import java.util.*;
class Stringfun
{
        String str;
        char ch;
        int v=0,c=0,up;
        Scanner sc=new Scanner(System.in);
        void input()
        {
        System.out.println("Input a String")          ;
        str=sc.next();
        }
        void words()
        {
        int l=str.length();

        for(int i=0;i<l;i++)
```

```
                {
                        char ch=str.charAt(i);


if(ch=='a'||ch=='A'||ch=='e'||ch=='E'||ch=='i'||ch=='I'||ch=='o'||ch=='O'||ch=='u'||ch=='U')
                                {

                                        v++;
                                }
                                        else if(Character.isUpperCase(ch))
                                        {
                                                up++;
                                        }


                }

                System.out.println("Total no of vowels="+v);
                System.out.println("Total no of words="+l);
                System.out.println("Total no of uppercases="+up);

        }
        void palindrome()
        {
                int len=str.length();
                String r="";
                for(int i=len-1;i>=0;i--)
                {
                        r=r+str.charAt(i);
                }


                        System.out.println("Reverse string="+r);

        }


public static void main(String args[])
{
        Stringfun ob=new Stringfun();
        ob.input();
        ob.words();
        ob.palindrome();
}
```

}

11.A superclass Detail has been defined to store the details of a customer.Define a subclass Bill to compute the monthly telephone charge of the customer as per the chart given below:

Number of calls	Rate
1-100	Only rental charge
101-200	60 paisa per call+rental charge
201 -300	80paisa per call+rental charge
Above 300	1 rupee per call+rental charge

Detail of both the classes are given below

Class name	**Detail**
Data members/Instance variables	
name	To store the name of the customer
address	To store address of the customer
telno	To store the phone number of customer
rent	To store the monthly rental charge
Member functions	
Detail(……)	Parameterized constructor to assign values to data members
void show()	To display the details of the customer
Class name	**Bill**
Data members/Instance variables	
n	To store the number of calls
Amt	To store the amount to be paid by customer
Member functions	
Bill(……)	Parameterized constructor to assign values to data members of both classes and to initialize amt=0.0
void cal()	Calculate the monthly telephone charge as per the chart given above
void show()	Display the details of the customer and amount to be paid

Ans.

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

public class Detail {
        String name,address,telno;
        double rent;
```

```
    public Detail(String n,String add,String no,double r)
        {
            name=n;
            address=add;
            telno=no;
            rent=r;
    }
    void show()
    {
        System.out.println("Name="+name);
        System.out.println("Address="+address);
        System.out.println("Telephone no="+telno);
        System.out.println("Rent="+rent);

    }


}
public class Bill extends Detail {
        int n;
        double amt;


    public Bill(String na,String a,String t,double re,int c)
        {
            super(na,a,t,re);
            n=c;
    }
    void cal()
    {
        if(n>=1&&n<=100)
            amt=rent;
            else if(n>=101&&n<=200)
                amt=rent+0.60;
                else if(n>=201&&n<=300)
                    amt=rent+0.80;
                    else if(n>300)
                    amt=rent+1.00;
    }
void show()
{
        super.show();
        System.out.println("Amount to be paid="+amt+"Rs");
```

```
}
public static void main(String args[])
{
        Bill ob=new Bill("smriti","kokar","8210549565",5.0,200);
        ob.cal();
        ob.show();
}


}
```

12.A superclass Record has been defined to store the names and ranks of 50 students.Define a subclass rank to find the highest rank alongwith the name.The details of both classes are given below:

Class name	**Record**
Data members/Instance variables	
n[]	To store the names of students
m[]	To store the ranks of students
size	To store the number of students
Member functions	
Record(int cap)	Constructor to initialize data members
void readarray()	To store the names and rank
void display()	Display the names and the corresponding ranks
class name	**Highest**
Data members/Instance variable	
ind	Integer to store the index of the topmost rank
Member functions	
Highest()	Parameterized Constructor to initialize the data members of both the classes.
void find()	Finds the index of the student obtaining the highest mark and assign it to 'ind'
void display()	Display the array elements along with the names and marks of the students who have obtained the highest mark.

Ans.

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

public class record {

   String n[];
   int m[];
```

```
int size;
public record(int cap)
{
	size=cap;
	n=new String[size];
	m=new int[size];
}
void readarray()
{
	Scanner sc=new Scanner(System.in);
	System.out.println("enter the arrays value");
	for(int i=0;i<size;i++)
	{
			System.out.println("Enter store name");
			n[i]=sc.next();
				System.out.println("Enter store marks");
				m[i]=sc.nextInt();}}
				void display()
				{
				System.out.println("printing the arrays value");
				System.out.println("Name\t\t Marks");
				for(int i=0;i<size;i++)
				{
					System.out.println(n[i]+"\t\t"+m[i]);
				}
				}
}public class highest extends record
{
	int ind;
	public highest(int size,int ind)
	{
		super(size);
		this.ind=ind;
	}
	void find()
	{
		int max=m[0];
		int l=-1;
		for(int i=0;i<m.length;i++)
		{
			if(m[i]>max)
			{
				max=m[i];
				l=i;
```

```
                    }
                }
                ind=l;
        }
        void display()
        {
                super.display();
                this.find();
                System.out.println("Highest marks obtained by the student");
                System.out.println("name of the student"+n[ind]);
                System.out.println("marks of the student"+m[ind]);
        }
        public static void main(String args[])
        {
                highest dbk=new highest(5,0);
                dbk.readarray();
                dbk.display();
        }
  }
```

13.Stack is a kind of data structure which can elements with the restriction that an element can be added or removed from the top only.
The details of the class Stack is given below:

Class name	**Stackop**
Data members/Instance variables	
Stack[]	The array to hold the names
N	The maximum capacity of the string array
Top	The index of the topmost element of the stack
Member functions	
Stackop()	Default constructor
Stackop(int size)	Parameterized constructor to initialize N=size and top=-1
void push(int data)	To push a name into the stack is full,display the message “Overflow”
void pop()	Removes an integer from the top of the stack and returns it,if the stack is empty, display the message “Underflow”
void display()	Display the elements of the stack

Ans.

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


public class stackop {
```

```
    int stack[];
    int top;
    int N;

public stackop()
    {
        N=0;
        top=-1;
        stack=new int[N];
}
public stackop(int size)
{
    N=size;
    top=-1;
    stack=new int[N];
}
public void push(int data)
{
    if(top==N)
    {
        System.out.print("Stack is full overflow error....");
    }
    else
    {
        top++;
        stack[top]=data;
    }
    System.out.println("New data pushed:"+data);

}
public void pop()
{
    int data;
    if(top==-1)
    {
        System.out.println("Stack is Empty. Underflow Error");
    }
    else
    {
        data=stack[top];
        System.out.println("Data deleted from the stack:"+data);
        top--;
    }
}
```

```
public void display()
{
    if(top==-1)
        System.out.println("Stack is empty");
        else
        {
            System.out.println("Stack values printing:");
                for(int i=0;i<=top;i++)
                {
                    System.out.print(stack[i]+"--------->");
                }
                System.out.println(".....");
        }
}
public static void main(String args[])
{
    stackop ob=new stackop(5);
    ob.push(5);
    ob.push(55);
    ob.push(30);
    ob.display();
    ob.pop();
    ob.display();
    ob.push(74);
    ob.display();
    ob.pop();
    ob.display();



}


}
```

14.Queue is an entity which can hold a maximum of 50 integers.The queue enables the users to add integers from the rear and remove integers from the front.Define a class Queue with the following details.

Class Name	**Product**
Data members/instance variables	
Q[]	Array to hold the integer elements
size	Store the size of the aray
front	To point the index of the front
rear	To point the index of the rear
Member functions	

Queue(int m)	Constructor to initialize the data size=m,front=0,rear=0
void insert(int data)	To add integer from the rear if possible else display the message “Over Flow”
void delete()	Returns elements from front if present, otherwise displays the message “Under Flow”
void display()	Display the array elements

Ans.

```
import java.io.*;
import java.util.*;
public class Queue
 {
        int Q[];
        int size;
        int front,rear;

   public Queue(int m)
         {
                size=m;
                front=0;
                rear=0;
                Q=new int[20];
   }
   void insert(int data)
   {
        if(rear==size-1)
        {
                System.out.println("Over Flow");
                return;
        }
        else if(rear==0)
        {
                front=1;
                rear=1;
        }
        else
                rear++;
                Q[rear]=data;
                System.out.println("Data inserted="+data);
   }
   void delete()
   {
        int data;
```

```
        if(front==0)
        {
                System.out.println("Queue is Empty:");
                return;
        }
        else
        {
                data=Q[front];
                if(front==rear)
                {
                        front=0;
                        rear=0;
                }
                else
                        front++;
                        System.out.println("Data deleted:"+data);
        }
}
        void display()
        {
                System.out.println("Queue data list is:");
                for(int i=front;i<=rear;i++)
                {
                        System.out.print(Q[i]+" ");
                }
                System.out.println();
        }
        public static void main(String args[])
        {
                Queue ob=new Queue(5);
                ob.insert(5);
                ob.display();
                ob.insert(34);
                ob.display();
                ob.insert(21);
                ob.display();
                        ob.insert(52);
                                ob.display();
                                ob.insert(17);
                                ob.display();
                                ob.delete();
                                        ob.display();
                                                ob.delete();
                                                ob.display();
```

```
        }
    }
```

15.A doubly queue is a linear data structure which enables the user to add and remove integers from either ends,i.e. from front or rear. Define a class Dequeue with the following details:

Class name	Dequeue
Data members/Instance variables	
arr[]	Array to hold up to 50 integers
front	To print the index of front end
rear	To print the index of rear end
Member functions	
Dequeue(int m)	Constructor to initialize the data members
void addfront(int data	To add integers from the front if possible else display the message “OverFlow from the Front”
Void addrear(int data)	To add integers from the rear if possible else display the message “OverFlow from the rear”
int popfront()	Returns element from front,if possible otherwise returns -9999
int poprear()	Returns element from rear ,if possible otherwise returns -9999

Ans.

```
import java.io.*;
import java.util.*;
public class Dequeue {
        int arr[];
        int front,rear;
        int size;

    public Dequeue(int m)
         {
                arr=new int[50];
                front=-1;
                rear=-1;
    }
    public void addfront(int data)
    {
        if(front==-1)
        {
                front=0;
```

```
                rear=0;
                arr[front]=data;
        }
        else if(front>0&&front<arr.length)
                arr[++front]=data;
                else
                        System.out.println("Can not be inserted from front.....");
}
public void addrear(int data)
{
        if(rear==-1)
        {
                front=0;
                rear=0;
                arr[rear]=data;
        }
        else if(rear+1<arr.length)
                arr[++rear]=data;
                else
                        System.out.println("Over Flow");
}
public int popfront()
{
        int element;
        if(front==-1)
                return -9999;
                else
                {
                        element=arr[front];
                        if(front==rear)
                                front=rear-1;
                                else
                                        front++;
                                        return element;
                }
}
public int poprear()
{
        int element;
        if(front==-1)
                return -9999;
                else
                {
                        element=arr[rear];
```

```
                    if(front==rear)
                        front=rear-1;
                        else
                            rear--;
                            return element;
                }
    }
    void display()
    {
        System.out.println("Printing values of Dequeue...");
        for(int i=front;i<=rear;i++)
        {
            System.out.println(arr[i]+" ");
        }
        System.out.println();
    }
    public static void main(String args[])
    {
        Dequeue ob=new Dequeue(20);
        ob.addfront(44);
        ob.addfront(32);
        ob.addrear(82);
        ob.addrear(60);
        ob.display();
        ob.popfront();
        ob.display();
        ob.poprear();
        ob.display();
    }


}
```

Note:
Project Work must include:

a) **Front Page**
b) **Introduction**
c) **Acknowledgement**
d) **Index**
e) **All project work program with questions. The program must include some comments and variable descriptions.**
f) **Bibliography**