

1) Bank Rate of Interest

```
interface Bank{
    abstract float rateOfInterest();
}
class SBI implements Bank{
    public float rateOfInterest(){
        return 9.15f;
    }
}
class PNB implements Bank{
    public float rateOfInterest(){
        return 9.7f;
    }
}
class Main{
    public static void main(String[] args){
        Bank b=new SBI();
        System.out.println("SBI Rate of Interest:"+b.rateOfInterest());
        PNB b1=new PNB();
        System.out.println("PNB Rate of Interest:"+b1.rateOfInterest());
    }
}
```

O/P:

SBI Rate of Interest:9.15

PNB Rate of Interest:9.7

2) Define an interface “IntOperations” with methods to check whether an integer is positive_negative, even_odd, prime and operations like factorial and sum of digits. Define a class MyNumber having one private int data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value (Use this). Implement the above interface. Create an object in main.

```
import java.util.*;
interface IntOperations{
    abstract void positiveNegative();
    abstract void evenOdd();
    abstract void prime();
    abstract void factorial();
    abstract void sumDigit();
}
class MyNumber implements IntOperations{
    private int n;
    MyNumber(){
        this.n=0;
    }
}
```

```

MyNumber(int n){
    this.n=n;
}
public void positiveNegative(){
    if(n>0)
        System.out.println("Number is Positive");
    else if(n<0)
        System.out.println("Number is Negative");
    else
        System.out.println("Number is Zero");
}
public void evenOdd(){
    if(n%2==0)
        System.out.println("Number is Even");
    else
        System.out.println("Number is Odd");
}
public void prime(){
    int count=0;
    for(int i=1;i<=n;i++){
        if(n%i==0){
            count++;
        }
    }
    if(count==2)
        System.out.println("Number is Prime");
    else
        System.out.println("Number is not Prime");
}
public void factorial(){
    int f1=1;
    for(int i=n;i>=1;i--){
        f1=f1*i;
    }
    System.out.println("Factorial="+f1);
}
public void sumDigit(){
    int sum=0;
    while(n>0){
        int digit=n%10;
        sum=sum+digit;
        n/=10;
    }
    System.out.println("Sum of Digit="+sum);
}

```

```

    }
}
public class Main
{
    public static void main(String[] args) {
        int n,i;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number");
        n=sc.nextInt();
        MyNumber mn=new MyNumber(n);
        mn.positiveNegative();
        mn.evenOdd();
        mn.prime();
        mn.factorial();
        mn.sumDigit();

    }
}

```

O/P:

Enter a number

11 2 1

Number is Positive

Number is Odd

Number is Prime

Factorial=39916800

Sum of Digit=2

3) Define an Interface Shape with abstract method area(). Write a java program to calculate an area of Circle and Sphere.(use final keyword)

```

import java.util.*;
interface Shape{
    final double pi=3.14;
    abstract void area();
}
class Circle implements Shape{
    Double r,A;
    Circle(double r){
        this.r=r;
    }
    public void area(){
        A=pi*r*r;
        System.out.println("Area of circle="+A);
    }
}

```

```

class Sphere implements Shape{
    Double r,A;
    Sphere(double r){
        this.r=r;
    }
    public void area(){
        A=4*pi*r*r;
        System.out.println("Area of Sphere="+A);
    }
}

public class Main{
    public static void main(String[] args){
        double r;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter value of r");
        r=sc.nextDouble();
        Circle c=new Circle(r);
        c.area();
        Sphere s=new Sphere(r);
        s.area();
    }
}

```

O/P:

Enter value of r

12

Area of circle=452.15999999999997

Area of Sphere=1808.6399999999999

4) Define an interface “Operation” which has methods area(),volume().Define a constant PI having a value 3.142.Create a class cylinder which implements this interface (members – radius, height) Create n object and calculate the area and volume.

```

import java.util.*;
interface Operation{
    final double PI=3.142;
    abstract void area();
    abstract void volume();
}

class Cylinder implements Operation{
    double r,h,A,V;
    Cylinder(double r,double h){
        this.r=r;
        this.h=h;
    }
    public void area(){

```

```

        A=2*PI*r*h+2*PI*r*r;
        System.out.println("Area of Cylinder="+A);
    }
    public void volume(){
        V=PI*r*r*h;
        System.out.println("Volume of Cylinder="+V);
    }
}

public class Main{
    public static void main(String[] args){
        int n,i;
        double r,h;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter array size");
        n=sc.nextInt();
        Cylinder[] c=new Cylinder[n];
        for(i=0;i<n;i++){
            System.out.println("Enter value of r and h");
            r=sc.nextDouble();
            h=sc.nextDouble();
            c[i]=new Cylinder(r,h);
            c[i].area();
            c[i].volume();
        }
    }
}

```

O/P:

Enter array size

2

Enter value of r and h

4

2

Area of Cylinder=150.816

Volume of Cylinder=100.544

Enter value of r and h

7

9

Area of Cylinder=703.808

Volume of Cylinder=1385.622

5) Write a program using market interface, create a class product(product_id, product_name, product_cost, product_quantity) define a default and parameterized constructor. Create objects of class product and display the contents of each object.
import java.util.*;

```

interface Marker{
    abstract void display();
}
class Product implements Marker{
    int product_id,product_quantity;
    String product_name;
    double product_cost;
    Product(){
        product_id=0;
        product_name="";
        product_cost=0.0;
        product_quantity=0;
    }
    Product(int product_id,String product_name,double product_cost,int product_quantity){
        this.product_id=product_id;
        this.product_name=product_name;
        this.product_cost=product_cost;
        this.product_quantity=product_quantity;
    }
    public void display(){
        System.out.println("Product id:"+product_id);
        System.out.println("Product name:"+product_name);
        System.out.println("Product cost:"+product_cost);
        System.out.println("Product quantity:"+product_quantity);
        System.out.println();
    }
}
public class Main{
    public static void main(String[] args){
        int n,i;
        int product_id,product_quantity;
        String product_name;
        double product_cost;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter array size");
        n=sc.nextInt();
        Product[] p=new Product[n];
        for(i=0;i<n;i++){
            System.out.println("Enter Product id,name,cost,quantity");
            product_id=sc.nextInt();
            product_name=sc.next();
            product_cost=sc.nextDouble();
            product_quantity=sc.nextInt();
            p[i]=new Product(product_id,product_name,product_cost,product_quantity);
        }
    }
}

```

```

        p[i].display();
    }
}

```

O/P:

Enter array size

2

Enter Product id,name,cost,quantity

1

Laptop

43999

7

Product id:1

Product name:Laptop

Product cost:43999.0

Product quantity:7

Enter Product id,name,cost,quantity

2

Mobile

10999

4

Product id:2

Product name:Mobile

Product cost:10999.0

Product quantity:4

6) Write a program to find the cube of a given number using the function interface.

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

```
interface Cube {
```

```
    abstract void cube1();
```

```
}
```

```
class CalculateCube implements Cube {
```

```
    int n;
```

```
    CalculateCube(int n){
```

```
        this.n=n;
```

```
    }
```

```
    public void cube1() {
```

```
        int result = n * n * n;
```

```
        System.out.println("The cube of " + n + " is: " + result);
```

```
    }
```

```
}
```

```
public class Main {
```

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

```

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number:");
        int number = sc.nextInt();
        CalculateCube calCube = new CalculateCube(number);
        calCube.cube1();
        sc.close();
    }
}

```

O/P:

Enter a number:

2

The cube of 2 is: 8

7) Define an interface “StackOperations” which declares methods for a static stack. Define a class “MyStack” which contains an array and top as data members and implements the above interface. Initialize the stack using a constructor. Write a menu driven program to perform operations on a stack object.

```

import java.util.*;
interface StackOperations{
    void push();
    void pop();
    void peek();
}

class stack implements StackOperations{
    Scanner sc=new Scanner(System.in);
    int top=2;
    int flag=1;
    int []a=new int[top];
    stack(){
        System.out.println("Enter "+top+" Values");
        for(int i=0;i<top;i++)
            a[i]=sc.nextInt();
    }
    public void push() {
        if(top<0 || flag==0) {
            top++;
            a=new int[top];
            int n=a.length;
            n--;
            System.out.println("Enter Element");
            a[n]=sc.nextInt();
            flag=1;
        }
    }
}

```



```

    }
    else {
        int i;
        int []b=new int[top];
        for(i=0;i<top;i++)
            b[i]=a[i];
        int n=a.length;
        top++;
        a=new int[top];
        for(i=0;i<top-1;i++)
            a[i]=b[i];
        System.out.println("Enter Element");
        a[n]=sc.nextInt();
    }
}
public void pop() {
    if(top<0 || flag==0)
        System.out.println("Stack is Empty");
    else {
        int i;
        int n=a.length;
        n--;
        System.out.println("Poped Element "+a[n]);
        int b[]=new int[top];
        for(i=0;i<top;i++)
            b[i]=a[i];
        top--;
        if(top>0) {
            a=new int[top];
            for(i=0;i<top;i++)
                a[i]=b[i];
        }
        else if(top==0) {
            flag=0;
        }
    }
}
public void peek() {
    if(top<0 || flag==0)
        System.out.println("Stack is Empty");
    else {
        int n=a.length;
        n--;
        System.out.println("Top Element "+a[n]);
    }
}

```

```

    }
    }
}
public class Main{
    public static void main(String[] args){
        stack s=new stack();
        int ch;
        Scanner sc=new Scanner(System.in);
        do
        {
            System.out.println("1.Push\n2.Peek\n3.Pop\n4.Exit");
            ch=sc.nextInt();
            switch(ch) {
                case 1: s.push();
                    break;
                case 2: s.peek();
                    break;
                case 3:s.pop();
                    break;
                case 4:System.exit(1);
                    break;
                default:System.out.println("Enter valid option");
                    break;
            }
        }while(ch!=4);
    }
}

```

8) Create an interface “CreditCardInterface” with methods to viewCreditAmount, viewPin, changePin, useCard and payBalance. Create a class Customer (name, card number, pin, creditAmount – initialized to 0). Implement methods viewCreditAmount, viewPin, changePin and payBalance of the interface. From Customer, create classes RegularCardHolder (maxCreditLimit) and GoldCardHolder (String specialPrivileges) and define the remaining methods of the interface. Create n objects of the RegularCardHolder and GoldCardHolder classes and write a menu driven program to perform the following actions 1. Use Card 2. Pay Balance 3. Change Pin

```

import java.util.*;
interface card{
    void viewAmt();
    void viewPin();
    void changePin();
}

class customer implements card{

```

```

Scanner sc=new Scanner(System.in);
String name;
int pin=1234,cardNo;
double amt=0;
public void viewAmt() {
    System.out.println("Amount "+amt);
}
public void viewPin() {
    System.out.println("Current pin "+pin);
}
public void changePin() {
    System.out.println("Enter new pin ");
    int npin=sc.nextInt();
    if(npin==pin)
        System.out.println("New pin cannot be same as old pin.");
    else
        pin=npin;
}
}

```

```

class regular extends customer{
    double maxAmt=250000;
    regular(){
        amt=maxAmt;
    }
    public void use() {
        System.out.println("Enter amount ");
        double useAmt=sc.nextDouble();
        if(useAmt<=amt) {
            amt-=useAmt;
            System.out.println("Transaction successfull\nBalance "+amt);
        }
        else
            System.out.println("Insufficient balance");
    }
    public void pay() {
        double pay=maxAmt-amt;
        if(pay>=0) {
            System.out.println("Pay\n\t1.Total Amount due "+pay+"\t2.Current Amount due "+(pay*0.50));
            int ch=sc.nextInt();
            switch(ch) {
                case 1:
                    amt+=pay;
                    System.out.println("Transaction successfull\nBalance "+amt);

```

```

        break;
    case 2:
        amt+=(pay*0.50);
        System.out.println("Transaction successfull\nBalance "+amt);
        break;
    default:
        System.out.println("You enter invalid option. Transaction cancelled.");
        break;
    }
}
else
    System.out.println("Your Credit Card has no pending payments.");
}
}

```

```

class gold extends customer{
    String special="your max Limit is 500000";
    double maxAmt=500000;
    gold(){
        amt=maxAmt;
    }
    public void use() {
        System.out.println("Enter amount ");
        double useAmt=sc.nextDouble();
        if(useAmt<=amt) {
            amt-=useAmt;
            System.out.println("Transaction successfull\nBalance "+amt);
        }
        else
            System.out.println("Insufficient balance");
    }
    public void pay() {
        double pay=maxAmt-amt;
        if(pay>=0) {
            System.out.println("Pay\n\t1.Total Amount due "+pay+"\t2.Current Amount due "+(pay*0.50));
            int ch=sc.nextInt();
            switch(ch) {
                case 1:
                    amt+=pay;
                    System.out.println("Transaction successfull\nBalance "+amt);
                    break;
                case 2:
                    amt+=(pay*0.50);
                    System.out.println("Transaction successfull\nBalance "+amt);

```

```

        break;
    default:
        System.out.println("You enter invalid option. Transaction cancelled.");
        break;
    }
}
else
    System.out.println("Your Credit Card has no pending payments.");
}
}
public class Main{
    public static void main(String[] args){
        int i,ch;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter n");
        int n=sc.nextInt();
        regular r[]=new regular[n];
        gold g[]=new gold[n];
        for(i=0;i<n;i++) {
            r[i]=new regular();
            g[i]=new gold();
        }
        do {
            System.out.println("1.Regular user\n2.Gold user");
            ch=sc.nextInt();
            if(ch==1) {
                System.out.println("Out of "+n+" user which one are you ");
                int user=sc.nextInt();
                user--;
                System.out.println("Enter pin ");
                int pin=sc.nextInt();
                if(r[user].pin==pin) {
                    System.out.println("1.Use card\n2.Pay Balance.\n3Change Pin");
                    int choice=sc.nextInt();
                    switch(choice) {
                        case 1:
                            r[user].use();
                            break;
                        case 2:
                            r[user].pay();
                            break;
                        case 3:
                            r[user].changePin();
                            break;
                    }
                }
            }
        } while (ch != 2);
    }
}

```

```

        default:
            System.out.println("Incorrect option");
        }
    }
    else
        System.out.println("Wrong pin");
}
else if(ch==2) {
    System.out.println("Out of "+n+" user which one are you ");
    int user=sc.nextInt();
    user--;
    System.out.println("Enter pin ");
    int pin=sc.nextInt();
    if(g[user].pin==pin) {
        System.out.println("Your speical privilege is, "+g[user].special);
        System.out.println("1.Use card\n2.Pay Balance.\n3Change Pin");
        int choice=sc.nextInt();
        switch(choice) {
            case 1:
                g[user].use();
                break;
            case 2:
                g[user].pay();
                break;
            case 3:
                g[user].changePin();
                break;
            default:
                System.out.println("Incorrect option");
        }
    }
    else
        System.out.println("Wrong pin");
}
else if(ch==3) {

}
else
    System.out.println("Invalid option");
}while(ch!=3);
}
}

```

9) Define an interface “QueueOperations” which declares methods for a static queue. Define a class “MyQueue” which contains an array and front and rear as data members and implements the above interface. Initialize the queue using a constructor. Write a menu driven program to perform operations on a queue object.

```
import java.util.*;
interface que{
    void push();
    void remove();
    void peek();
}

class MyQueue implements que{
    Scanner sc=new Scanner(System.in);
    int front,rear,i;
    int a[];
    MyQueue(){
        front=0;
        rear=1;
        this.a=new int[rear+1];
        System.out.println("Enter "+(rear+1)+" values");
        for(i=front;i<=rear;i++)
            a[i]=sc.nextInt();
    }

    public void push() {
        int b[]=new int[rear+1];
        for(i=front;i<=rear;i++)
            b[i]=a[i];
        rear++;
        this.a=new int[rear+1];
        System.out.println("Enter Element ");
        a[rear]=sc.nextInt();
        for(i=front;i<rear;i++)
            a[i]=b[i];
        for(i=front;i<=rear;i++)
            System.out.println(a[i]);
    }

    public void remove() {
        if(rear<0)
            System.out.println("Queue is empty");
        else {
            System.out.println("Removed Element "+a[front]);
            int b[]=new int[a.length];
            for(i=0;i<a.length;i++)
```

```

        b[i]=a[i];
        rear--;
        this.a=new int[rear+1];
        for(i=front;i<b.length-1;i++)
            a[i]=b[i+1];
    }
}
public void peek() {
    if(rear<0)
        System.out.println("Queue is empty");
    else
        System.out.println("Front Element "+a[front]);
}
}
public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        MyQueue q=new MyQueue();
        int ch;
        do {
            System.out.println("1.Push\n2.Peek\n3.Pop\n4.Exit");
            ch=sc.nextInt();
            switch(ch) {
                case 1:
                    q.push();
                    break;
                case 2:
                    q.peek();
                    break;
                case 3:
                    q.remove();
                    break;
                case 4:
                    System.exit(1);
                    break;
                default:
                    System.out.println("Enter valid option");
                    break;
            }
        }while(ch!=4);
    }
}

```