

1. Grocery Problem

```
class Grocery
{
String c_name;
String c_ph;
double total;

Grocery(String c_name, String c_ph){
this.c_name= c_name;
this.c_ph = c_ph;
}

void calc(double q_dal,double q_pulses, double q_sugar) {
total = q_dal*100+q_pulses*80+q_sugar*50;
}

void display()
{
System.out.println("Name "+" "+"Phone number "+"
"+"Total");
System.out.println(c_name+" "+"c_ph+" "+total);
System.out.println();
}

}
class GDemo
{
public static void main(String[]args)
{
Grocery g1 = new Grocery("Rama","8060302010");
Grocery g2 = new Grocery("Shama","7689632510");
Grocery g3 = new Grocery("Bhama", "9632587412");
g1.calc(2, 2, 1);
g1.display();
g2.calc(3, 5, 2);
g2.display();
g3.calc(1, 1, 0.5);
g3.display();
}}
```

```
C:\Users\sonal\OneDrive\Desktop\sonaljava>java GDemo
Name Phone number Total
Rama 8060302010 410.0

Name Phone number Total
Shama 7689632510 800.0

Name Phone number Total
Bhama 9632587412 205.0
```

2. Method Overloading

```
class Overload
{
void print(int n) {
int sum = 0;
for(int i=1; i<=n;i++) {
sum = sum+i;
}
System.out.println("Sum of "+n+" natural numbers is
"+sum);
}
void print(int m, int n) {
System.out.println("Prime numbers in the range are ");
for(int i=m;i<=n;i++) {
int flag=0;
for(int j=2;j<=i/2;j++) {
if(i%j == 0) {
flag = 1;
break;
}
}
if(flag==0)
System.out.println(i);
}
}
}
class OverloadDemo {
public static void main(String[] args) {
Overload o = new Overload();
o.print(5);
o.print(7,13);
}
}
```

```
C:\Users\sonal\OneDrive\Desktop\sonaljava>java OverloadDemo
Sum of 5 natural numbers is 15
Prime numbers in the range are
7
11
13
```

3. Quadratic Equation

```
import java.util.Scanner;
class Quad {
int a, b, c;
double root1, root2, d;
Scanner s = new Scanner(System.in);
void input()
{
System.out.println("Quadratic equation is in the form : ax^2
+ bx + c");
System.out.print("Enter a:");
a = s.nextInt();
System.out.print("Enter b:");
b = s.nextInt();
System.out.print("Enter c:");
c = s.nextInt();
}
void discriminant() {
d= (b*b)-(4*a*c);
}
void calculateRoots() {
if(d>0)
{
System.out.println("Roots are real and unequal");
root1 = ( - b + Math.sqrt(d))/(2*a);
root2 = (-b - Math.sqrt(d))/(2*a);
System.out.println("First root is:"+root1);
System.out.println("Second root is:"+root2);
}
else if(d == 0)
{
System.out.println("Roots are real and equal");
root1 = (-b+Math.sqrt(d))/(2*a);
System.out.println("Root:"+root1);
}
else
{
System.out.println("No real solutions. Roots are
imaginary");
double real = -b / (2 * a);
double imaginary = Math.sqrt(-d) / (2 * a);
System.out.println("The equation has two complex roots: " +
real + " + " + imaginary + "i and "
+ real + " - " + imaginary + "i");
}
}
```

```

}
}
class Main {
public static void main(String[] args) {
Quad q= new Quad();
q.input();
q.discriminant();
q.calculateRoots();
}
}

```

```

C:\Users\sonal\OneDrive\Desktop\sonaljava>java Main
Quadratic equation is in the form :  $ax^2 + bx + c$ 
Enter a:2
Enter b:3
Enter c:3
No real solutions. Roots are imaginary
The equation has two complex roots:  $0.0 + 0.9682458365518543i$  and  $0.0 - 0.9682458365518543i$ 

```

```

C:\Users\sonal\OneDrive\Desktop\sonaljava>java Main
Quadratic equation is in the form :  $ax^2 + bx + c$ 
Enter a:1
Enter b:-7
Enter c:10
Roots are real and unequal
First root is:5.0
Second root is:2.0

```

```

C:\Users\sonal\OneDrive\Desktop\sonaljava>java Main
Quadratic equation is in the form :  $ax^2 + bx + c$ 
Enter a:1
Enter b:-4
Enter c:4
Roots are real and equal
Root:2.0

```

4. Percentage calculation

```
import java.util.Scanner;
class Student
{
    String USN;
    String Name;
    double per;
    double marks[]=new double[6];

    Scanner sc= new Scanner(System.in);

    void accept()
    {
        System.out.println("Enter the usn");
        USN=sc.nextLine();
        System.out.println("Enter the name");
        Name=sc.nextLine();
        System.out.println("Enter the marks");
        for(int i=0;i<6;i++)
        {
            marks[i]=sc.nextDouble();
        }
    }

    void percentage() {
        double sum = 0;
        for (int i = 0; i < 6; i++) {
            sum += marks[i];
        }
        per = (sum / 600) * 100;
    }

    void display()
    {
        System.out.println(USN);
        System.out.println(Name);
        System.out.println(per);
    }
}

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

```
int n;  
Scanner sc = new Scanner(System.in);  
System.out.println("Enter the number of students");  
n = sc.nextInt();  
Student s[] = new Student[n];  
for(int i=0;i<n;i++)  
{  
s[i]=new Student();  
s[i].accept();  
s[i].percentage();  
s[i].display();  
}  
}  
}
```

```
C:\Users\sonal\OneDrive\Desktop\sonaljava>java Run  
Enter the number of students  
2  
Enter the usn  
1bm22cs286  
Enter the name  
sonal  
Enter the marks  
100 100 100 100 100 100  
1bm22cs286  
sonal  
100.0  
Enter the usn  
1bm22cs400  
Enter the name  
sonakshi  
Enter the marks  
100 80 80 90 50 60  
1bm22cs400  
sonakshi  
76.66666666666667
```

5. Book Class Object Program

```
import java.util.Scanner;
class Books {
String name;
String author;
int price;
int numPages;

Books(String name,String author,int price, int numPages)
{ this.name=name;
this.author=author;
this.price=price;
this.numPages=numPages;
}

public String toString()
{
String name,author,price,numPages;
name="Book Name:"+this.name+"\n";
author="Author Name:"+this.author+"\n";
price="Price:"+this.price+"\n";
numPages="Number of
Pages:"+this.numPages+"\n";
return name+author+price+numPages;
}
}

import java.util.Scanner;
class MainBook {
public static void main(String args[ s])
{
Scanner scan=new Scanner(System.in);
int n,price,numPages;
String name,author;
System.out.println("Enter no.of books:\n");
n=scan.nextInt();
Books b[ ]=new Books[n];
for(int i=0;i<n;i++)
{
System.out.println("Enter details of the book"+(i+1)+":");
System.out.println("Enter name of book:");
name=scan.next();
scan.nextLine();
```

```

System.out.println("Enter author name:");
author=scan.nextLine();
System.out.println("Enter price of book:");
price=scan.nextInt();
System.out.println("Enter no.of pages:");
numPages=scan.nextInt();
b[i]=new Books(name,author,price,numPages);
}
System.out.println("Book Details:");
System.out.println("Book Name\tAuthor\tPrice\tNo.of Pages");
for(int i=0;i<n;i++)
{
System.out.println(b[i].name+"\t"+b[i].author+"\t"+b[i].price+"\t"+b[i].numPages); }
}
}

```

```

C:\Users\sonal\OneDrive\Desktop\sonaljava>java MainBook
Enter no.of books:

2
Enter details of the book1:
Enter name of book:
jungle book
Enter author name:
rudyard kipling
Enter price of book:
500
Enter no.of pages:
40
Enter details of the book2:
Enter name of book:
akbar and birbal
Enter author name:
akbar
Enter price of book:
700
Enter no.of pages:
70
Book Details:
jungle  rudyard kipling 500      40
akbar   akbar    700      70

```


6. Abstract Class – Shape

```
abstract class Shape
```

```
{  
    double length;  
    double breadth;  
    double radius;  
    abstract void Printarea();  
}
```

```
class Rectangle extends Shape
```

```
{  
    Rectangle(double length,double breadth)  
    {  
        this.length=length;  
        this.breadth=breadth;  
    }  
    void Printarea()  
    {  
        double area;  
        area=length*breadth;  
        System.out.println(area);  
    }  
}
```

```
class Triangle extends Shape
```

```
{  
    Triangle(double length,double breadth)  
    {  
        this.length=length;  
        this.breadth=breadth;  
    }  
    void Printarea()  
    {  
        double area;  
        area=0.5*length*breadth;  
        System.out.println(area);  
    }  
}
```

```
class Circle extends Shape
```

```
{  
    Circle(double radius)  
    {  
        this.radius=radius;  
    }  
}
```

```
void Printarea()
{
double area;
area=3.14*radius*radius;
System.out.println(area);
}
}

class MainRun
{
public static void main(String[]args)
{
Rectangle r =new Rectangle(1,2);
Triangle t =new Triangle(2,5);
Circle c = new Circle(2);
Shape s;
s=r;
s.Printarea();
s=t;
t.Printarea();
s=c;
c.Printarea();
}
}
```

```
C:\Users\sonal\OneDrive\Desktop\sonaljava>java MainRun
2.0
5.0
12.56
```

7. Bank Account

```
import java.util.Scanner;

class Account {
    String cname;
    long accno;
    String type;
    double balance;

    Account(String cn, long ac, String t, double b) {
        cname = cn;
        accno = ac;
        type = t;
        balance = b;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. Current balance=" + balance);
    }

    void displayBalance() {
        System.out.println("Balance is: " + balance);
    }
}

class SavAcct extends Account {
    SavAcct(String cn, long ac, String t, double b) {
        super(cn, ac, t, b);
    }

    void computeAndDepositInterest(double rate, int time) {
        double interest = balance * Math.pow((1 + rate), time) - balance;
        deposit(interest);
    }

    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. Current balance=" + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}
```

```

class CurAcct extends Account {
    final double MIN_BALANCE = 1000; // Minimum balance for current account
    final double SERVICE_CHARGE = 50; // Service charge for falling below minimum
    balance

    CurAcct(String cn, long ac, String t, double b) {
        super(cn, ac, t, b);
    }

    void withdraw(double amount) {
        if (balance - amount >= MIN_BALANCE) {
            balance -= amount;
            System.out.println("Withdrawal successful. Current balance=" + balance);
        } else {
            System.out.println("Insufficient balance. Service charge of Rs. " +
SERVICE_CHARGE + " imposed.");
            balance -= SERVICE_CHARGE;
            System.out.println("Current balance after service charge: " + balance);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        boolean exit = false;
        while (!exit) {
            System.out.println("Enter Name:");
            String name = scan.nextLine();
            System.out.println("Enter Account Number:");
            long accNo = scan.nextLong();
            System.out.println("Enter account type: (Savings/Current)");
            String type = scan.next();
            System.out.println("Enter opening balance:");
            double balance = scan.nextDouble();

            if (type.equalsIgnoreCase("Savings")) {
                SavAcct savAcct = new SavAcct(name, accNo, type, balance);
                System.out.println("Savings Account created.");
                int choice;
                do {
                    System.out.println("1. Deposit\n2. Display Balance\n3. Compute and Deposit
Interest\n4. Withdraw\n5. Exit");
                    System.out.println("Enter your choice:");
                    choice = scan.nextInt();

```

```

switch (choice) {
    case 1:
        System.out.println("Enter deposit amount:");
        double depositAmount = scan.nextDouble();
        savAcct.deposit(depositAmount);
        break;
    case 2:
        savAcct.displayBalance();
        break;
    case 3:
        System.out.println("Enter rate of interest:");
        double rate = scan.nextDouble();
        System.out.println("Enter time period (in years):");
        int time = scan.nextInt();
        savAcct.computeAndDepositInterest(rate, time);
        break;
    case 4:
        System.out.println("Enter withdrawal amount:");
        double withdrawAmount = scan.nextDouble();
        savAcct.withdraw(withdrawAmount);
        break;
    case 5:
        exit = true;
        System.out.println("Exiting...");
        break;
    default:
        System.out.println("Invalid choice.");
}
} while (!exit);
} else if (type.equalsIgnoreCase("Current")) {
    CurAcct curAcct = new CurAcct(name, accNo, type, balance);
    System.out.println("Current Account created.");
    int choice;
    do {
        System.out.println("1. Deposit\n2. Display Balance\n3. Withdraw\n4. Exit");
        System.out.println("Enter your choice:");
        choice = scan.nextInt();
        switch (choice) {
            case 1:
                System.out.println("Enter deposit amount:");
                double depositAmount = scan.nextDouble();
                curAcct.deposit(depositAmount);
                break;
            case 2:
                curAcct.displayBalance();
                break;

```

```

        case 3:
            System.out.println("Enter withdrawal amount:");
            double withdrawAmount = scan.nextDouble();
            curAcct.withdraw(withdrawAmount);
            break;
        case 4:
            exit = true;
            System.out.println("Exiting...");
            break;
        default:
            System.out.println("Invalid choice.");
    }
    } while (!exit);
} else {
    System.out.println("Invalid account type.");
}
}
}
}

```

```

C:\Users\sonal\OneDrive\Desktop\sonaljava>java Bank
Enter Name:
sonal
Enter Account Number:
1234
Enter account type: (Savings/Current)
current
Enter opening balance:
3000
Current Account created.
1. Deposit
2. Display Balance
3. Withdraw
4. Exit
Enter your choice:
1
Enter deposit amount:
5000
Deposit successful. Current balance=8000.0

```

```
1. Deposit
2. Display Balance
3. Withdraw
4. Exit
Enter your choice:
2
Balance is: 8000.0
1. Deposit
2. Display Balance
3. Withdraw
4. Exit
Enter your choice:
3
Enter withdrawal amount:
4000
Withdrawal successful. Current balance=4000.0
1. Deposit
2. Display Balance
3. Withdraw
4. Exit
Enter your choice:
2
Balance is: 4000.0
1. Deposit
2. Display Balance
3. Withdraw
4. Exit
Enter your choice:
4
```

```
Exiting...
```

8. Package Program – CIE, SEE

```
//Sonal - 1BM22CS286
//Student Class package
CIE;
import java.util.Scanner;

public class Student { protected String
    usn = new String(); protected String
    name = new String(); protected int
    sem; public void inputStudentDetails()
    { Scanner scan=new
    Scanner(System.in);
    System.out.println("Enter name:");
    name=scan.next();
    System.out.println("Enter usn:");
    usn=scan.next();
    System.out.println("Enter sem:");
    sem=scan.nextInt();

} public void displayStudentDetails()
{
    System.out.println("Name:"+name);
    System.out.println("USN:"+usn);
    System.out.println("Sem:"+sem);
}
}

//Internals class package
CIE; import
java.util.Scanner;
public class Internals extends Student
{
    protected int marks[] = new int[5];
    public void inputCIEmarks()

    {
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter internal marks of 5 subjects:");
        for(int i=0;i<marks.length;i++)
        {
            marks[i]=scan.nextInt();
        }
    }
}
```



```

//External class package
SEE; import
CIE.Internals;
    Import
java.util.Scanner; public
class External extends
Internals { protected int
marks[];
protected int finalMarks[];

public External() {
    marks = new int[5];
    finalMarks = new int[5];
}

public void inputSEEmarks() { Scanner
s = new Scanner(System.in);
for(int i=0;i<5;i++) {
System.out.print("Subject "+(i+1)+" marks: ");
marks[i] = s.nextInt();
}
}

public void calculateFinalMarks() { for(int
i=0;i<5;i++)
finalMarks[i] = marks[i]/2 + super.marks[i];
}

public void displayFinalMarks() {
displayStudentDetails(); for(int
i=0;i<5;i++)
System.out.println("Subject " + (i+1) + ": " + finalMarks[i]);
}
}

//MainMarks
import SEE.External;

class MainMarks
{
public static void main(String args[])
{
int numOfStudents = 2;
External finalMarks[] = new External[numOfStudents];
for(int i=0;i<numOfStudents;i++)
{

```

```

finalMarks[i] = new External();
finalMarks[i].inputStudentDetails();
System.out.println("Enter CIE marks");
finalMarks[i].inputCIEmarks(); System.out.println("Enter
SEE marks"); finalMarks[i].inputSEEmarks();
}
System.out.println("Displaying data:\n"); for(int
i=0;i<numOfStudents;i++)
{
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
} //end of for loop
}
}

```

```

C:\Users\sonal\OneDrive\Desktop\main> java mainmarks
Enter name:
sonal
Enter usn:
1bm22cs286
Enter sem:
2
Enter CIE marks
Enter internal marks of 5 subjects:
35 35 35 35 35 35
Enter SEE marks
Subject 1 marks: 33
Subject 2 marks: 33
Subject 3 marks: 33
Subject 4 marks: 333
Subject 5 marks: 33
Enter name:
vRSH
Enter usn:
3535
Enter sem:
4
Enter CIE marks
Enter internal marks of 5 subjects:
33
33
33

```

```
33
33
Enter SEE marks
Subject 1 marks: 33
Subject 2 marks: 33
Subject 3 marks: 33
Subject 4 marks: 33
Subject 5 marks: 33
Displaying data:
```

```
Name:sonal
USN:1bm22cs286
Sem:2
Subject 1: 51
Subject 2: 51
Subject 3: 51
Subject 4: 201
Subject 5: 51
Name:vRSH
USN:3535
Sem:4
Subject 1: 49
Subject 2: 49
Subject 3: 49
Subject 4: 49
Subject 5: 49
```

9. Exception Handling – User Defined Exceptions

```
class WrongAge extends Exception {
    WrongAge() {
        super("Invalid age! Age cannot be negative.");
    }
}

class Father {
    int age;
    Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }
}

class Son extends Father {
    int sonAge;
    Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge();
        }
        this.sonAge = sonAge;
    }
}

public class ExceptionDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(25);
            System.out.println("Father's age: " + father.age);
            Son son = new Son(25, 50);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

```
C:\Users\sonal\OneDrive\Desktop\sonaljava>javac ExceptionDemo.java

C:\Users\sonal\OneDrive\Desktop\sonaljava>java ExceptionDemo
Father's age: 25
Exception caught: Invalid age! Age cannot be negative.
```

10. Threads

```
class DisplayBMS extends Thread {
    public void run() {
        while (true) {
            try {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // 10 seconds
            } catch (InterruptedException e) {
                System.out.println("exception caught");
            }
        }
    }
}
```

```
class DisplayCSE extends Thread {
    public void run() {
        while (true) {
            try {
                System.out.println("CSE");
                Thread.sleep(2000); // 2 seconds
            } catch (InterruptedException e) {
                System.out.println("exception caught");
            }
        }
    }
}
```

```
public class ThreadDemo {
    public static void main(String[] args) {
        DisplayBMS bmsThread = new DisplayBMS();
        DisplayCSE cseThread = new DisplayCSE();

        bmsThread.start();
        cseThread.start();
    }
}
```

```
C:\Users\sonal\OneDrive\Desktop\sonaljava>java ThreadDemo
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```