

## ① Grocery

class Grocery

{  
int qty\_dal;

int qty\_pulses;

int qty\_sugar;

Grocery ( int a, int b, int c )

{

qty\_dal = a;

qty\_pulses = b;

qty\_sugar = c; // for mid<sup>2</sup>) writing less code

}

void calc()

{

System.out.println ("Total cost is " +

(150 \* qty\_dal) + (18 \* qty\_pulses) + (50 \* qty\_sugar)

}

class shopping

{

public static void main( String args[] )

{

Grocery g1 = new Grocery(1,2,3);

Grocery g2 = new Grocery(1,2,3);

Grocery g3 = new Grocery(2,3,4);

g1.calc();

g2.calc();

g3.calc();

}

## Output

Total cost is 15036150

Total cost is 15036150

Total cost is 30054200

Q) WAP to overload the method print

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");

{

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 Overload Demo {  
    public static void main (String args)  
    {  
        Overload o = new Overload ();  
        o.print(5);  
        o.print(7, 13);  
    }  
}
```

Output  
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();  
        if (a == 0)  
        {  
            System.out.println ("Invalid Inputs");  
        }
```

```
void discriminant () {
```

$$d = (b * b) - (4 * a * c);$$

}

```
void calculateRoots () {
```

if ( $d > 0$ )

{

System.out.println ("Roots are real and unequal");

$$\text{root 1} = (-b + \sqrt{d}) / (2 * a);$$

$$\text{root 2} = (-b - \sqrt{d}) / (2 * a);$$

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");

$$\text{root 1} = (-b + \sqrt{d}) / (2 * a);$$

System.out.println ("Root :" + root1);

}

else

{

System.out.println ("No real solutions. Roots are  
Imaginary");

$$\text{double real} = -b / (2 * a);$$

$$\text{double imaginary} = \sqrt{-d} / (2 * a);$$

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();

( $a == 0 || b == 0 || c == 0$ )

~~if ( $a == 0$ ) {  
System.out.println ("a != 0");}~~

~~if ( $b == 0$ ) {  
System.out.println ("b != 0");}~~

~~if ( $c == 0$ ) {  
System.out.println ("c != 0");}~~

q. discriminant();  
q. calculateRoots();  
}

Output

Quadratic Equation is in the form :  $ax^2 + bx + c$

Enter a : 5

Enter b : 4

Enter c : 2

No real solutions. Roots are Imaginary

The equation has two complex roots :

$0.0 + 0.4898979i$  and  $0.0 - 0.4898979i$

roots not  
found option

11/24

```
1. import java.util.*;  
class Student  
{  
    String USN;  
    String name;  
    double marks[] = new double[6];  
    double P;  
    void details()  
{  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter USN");  
        USN = sc.next();  
        System.out.println ("Enter Name");  
        name = sc.next();  
        System.out.println ("Enter marks for 6 Sub");  
        for (int i = 0; i < 6; i++)  
        {  
            System.out.print ("Subject " + (i+1) + ": ");  
            marks[i] = Scanner.nextDouble();  
        }  
    }  
}
```

```
double double calculatePercentage()
```

```
{  
    double totalMarks = 0;  
    for
```

```
    double calc()  
{
```

```
        double sum = 0;  
        for (i = 0; i < 6; i++)  
    }
```

```
        sum = sum + marks[i];
```

```
    }  
    P = (sum / 600) * 100;
```

```
void display()
{
    System.out.println (USN);
    System.out.println (name);
    System.out.println (Per);
    for (i = 0; i < 6; i++)
    {
        System.out.println (marks[i]);
    }
    System.out.println (P);
}
```

### class Runner

```
public static void main (String [] args)
{
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the no. of students");
    int numStudents = scanner.nextInt();
    Student [] students = new Student [numStudents];
    for (int i = 0; i < numStudents; i++)
    {
        Students [i] = new Student ();
        System.out.println ("Enter details " + (i + 1) + ":");
        Students [i].accept () details;
    }
    System.out.println ("Student Details :");
    for (Student student : students)
    {
        student.display ();
    }
}
```

Output

Enter USN

IBM

Enter the number of students

2

Enter details

IBM22CS201

Enter USN

IBM22CS201

Enter Name

Ram

Enter marks for 6 sub

70

80

60

50

40

90

$$P = 65\%$$

IBM22CS201

Ram

70

80

60

50

40

90

P = 65%

Enter details

Enter USN

IBM22CS200

Enter Name

Shyam

Enter marks for 6 sub

70

80

65

25

35

40

$$P = 52.5$$

1BM22CS200

Shyam

90

80

65

25

35

40

~~52.5~~

## 2) Class Book

{

String name;

String author;

int price;

int numPages;

Book (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; + "\n";

```
class Main
```

{

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

{

```
        Scanner s = new Scanner (System.in),
```

```
        int n;
```

```
        String name;
```

```
        String author;
```

```
        int price;
```

```
        int numPages;
```

```
        System.out.println ("Enter the number of books:");
```

```
        n = s.nextInt();
```

```
        Books b[];
```

```
        b = new Books [n];
```

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

{

~~System.out.println ("Book");~~~~System.out.print ("Enter Name of book");~~~~name = s.next();~~~~System.out.print ("Enter author");~~~~author = s.next();~~~~System.out.print ("Enter price");~~~~price = s.nextInt();~~~~System.out.print ("Enter no. of pages");~~~~numPages = s.nextInt();~~~~b[i] = new Books (name, author, price,  
 numPages);~~

{

~~for (int i=0; i<n; i++)~~~~System.out.println (Book + b[i]));~~

Output

Enter the number of book : 2

Book 1

Enter the name of book : Jungle-Book

Enter the author of book : Rudyard Kipling

Enter the price of book : 1000

Enter the number of pages of book : 800

Book 2

Enter the name of book : Akbar and Birbal

Enter the author of book : Birbal

Enter the price of book : 900

Enter the pages of book : 400

Book 1

~~book name :~~

~~84~~  
~~87-24~~

abstract class Shape

{  
double l;

double w;

shape (double a, double b)

{  
l = a;  
w = b;

{  
abstract void printArea();

{  
class Rectangle extends Shape {

Rectangle (double a, double b)

{  
super (a, b);

void printArea()

{  
double area;  
area = l \* w;

System.out.println (area);

class Triangle extends Shape

{  
Triangle (double a, double b)

{  
super (a, b);

void printArea()

{  
double b;

b = 0.5 \* l \* w;

System.out.println (b);

class Circle extends Shape

{

Circle (double a, double b)

{

super (a, b);

{

void printArea ()

{

double c;

c = 3.14 \* l \* l;

System.out.println (c);

{

{

class Run {

public static void main (String [] args)

{

Rectangle r1 = new Rectangle (10, 20);

Triangle t1 = new Triangle (2, 4);

Circle c1 = new Circle (2, 0);

Shape s;

s = r1;

s.printArea ();

s = t1;

s.printArea ();

s = c1;

s.printArea ();

{

{

Output

200

4.0

12.56

✓  
19/1/24

2. `import java.util.Scanner;`

```
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;
```

```
Account(String customerName, int accountNumber,  
        String accountType, double balance)  
{
```

```
    this.customerName = customerName;  
    this.accountNumber = accountNumber;  
    this.accountType = accountType;  
    this.balance = balance;
```

}

```
public void deposit(double amount)  
{
```

```
    balance = balance + amount;
```

```
System.out.println(balance);
```

~~```
void displayBalance() {
```~~~~```
    System.out.println("Current Balance : " + balance);
```~~

}

```
class Current extends Account  
{
```

```
    double minBalance;  
    double serviceCharge;
```

CustAcct (String customerName, int accountNumber,  
double balance, double minBalance, double  
serviceCharge) {

super (customerName, accountNumber, balance);  
this.minBalance = minBalance;  
this.serviceCharge = serviceCharge;

{

void checkMinBalance()

{

if (balance < minBalance)

{

balance = balance - serviceCharge;

System.out.println ("Service charge imposed"  
+ balance);

{

{

void deposit (double amount)

{

super.deposit (amount)

checkMinBalance();

{

void withdraw (double amount)

{

balance = balance - amount;

System.out.println ("Withdrawal successful.  
Updated Balance : " + balance);

checkMinBalance();

{

{

class SavAcct extends Account {

double interestRate;

SavAcct (String customerName, int accountNumber,  
double balance, double interestRate) {

Super (customer Name, account Number, "Savings";  
balance);  
this. interestRate = interestRate;

3

void computeAndDepositInterest()

{

double interest = balance \* interestRate / 100;

balance = balance + interest;

System.out.println ("Interest deposited  
Updated balance :" + balance);

void deposit (double amount)

{

super.deposit(amount)

computeAndDepositInterest();

}

void withdraw (double amount) {

balance = balance - amount;

System.out.println ("Updated balance :" + balance);

3

3

class BankDemo

{

public static void main (String [] args)

Current Account = new CurrentAccount

("John Doe", 123456, 1000, 500, 10);

currentAccount.deposit(200);

currentAccount.displayBalance();

currentAccount.withdraw(300);

SavAcct savings Account = new SavAcct ("Jane Doe",  
789012, 1500, 5);

savings Account . deposit(300);

savings Account . display Balance();

savings Account . withdraw(100);

y

3

Q.P. Choose account type:

1. current

2. savings

Enter choice (1 or 2) :

Enter customer name : Sonal

Enter account number : 123

Enter initial balance : \$2000

Enter withdrawal amount : \$300

withdraw ~~not~~ successful . current Balance : \$1700

3 (Customer private saving  
new method)

3 (Customer private saving  
new method)

3 (Customer private saving  
new method)

withdraw always shows new method

## Packages

```
package CIE  
class Student  
{
```

```
private String usn;  
private String name;  
private int sem;
```

```
class Internals extends Student
```

{

```
int a[] = new int[20];
```

```
public Student (String usn, String name, int sem)
```

{

```
this.usn = usn;
```

```
this.name = name;
```

```
this.sem = sem;
```

}

```
public String getUSN() {
```

```
    return usn;
```

```
public String getName() {
```

```
    return name;
```

```
public int getSem() {
```

```
    return sem;
```

```
public class Internals extends Student
```

```
{ private int [] marks;
```

```
public Internals (String usn, String name, int sem,  
int [] marks) {
```

```
super (usn, name, sem);  
this.marks = marks;  
}  
public int [] getMarks () {  
    return marks;  
}  
}  
package SEE;  
public class External extends Student {  
    private int [] marks;  
    public External (String usn, String name, int sem,  
                    int [] marks) {  
        super (usn, name, sem);  
        this.marks = marks;  
    }  
    public int [] getMarks () {  
        return marks;  
    }  
}  
import CIE. Internals;  
import SEE. Internals;  
public class Main {  
    public static void main (String [] args) {  
        int n = Integer.parseInt (args [0]);  
        int [][] internalMarks = new int [n] [5];  
        int [][] externalMarks = new int [n] [5];  
    }  
}
```

Q1P

Student 1  
USN: 320  
Name: Sonal

Student 2  
USN: 321  
Name: son

for (int i = 0; i < n; i++)

String usn = args[1 + 5 \* i];

String name = args[2 + 5 \* i];

int sem = Integer.parseInt(args[3 + 5 \* i]);

for (j = 0; j < 5; j++)

internalMarks[i][j] = Integer.parseInt  
(args[4 + 5 \* i + j]);

for (int j = 0; j < 5; j++)

externalMarks[i][j] = Integer.parseInt  
(args[9 + 5 \* i + j]);

for (i = 0; i < n; i++)

int[] finalMarks = new int[5];

for (int j = 0; j < 5; j++)

finalMarks[j] = (internalMarks[i].internalMarks[j]  
+ externalMarks[i].externalMarks[j]) / 2;

System.out.println("Student " + (i + 1));  
internalMarks[i].display();  
externalMarks[i].display();

System.out.println("Final Marks: ");

for (int j = 0; j < 5; j++)

System.out.print("course" + (j + 1) + ":" +  
finalMarks[j]);

System.out.println();

(2)

Thread

```

class onethread extends Thread
{
    public void run()
    {
        try
        {
            for(int i=0; i<5; i++)
                System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
        }
        catch(InterruptedException e)
        {
            System.out.println("Catch");
        }
    }
}

```

## class twothread extends Thread

```

{
    public void run()
    {
        try
        {
            for(int i=0; i<5; i++)
                System.out.println("CSE");
            Thread.sleep(20000);
        }
        catch(InterruptedException e)
        {
            System.out.println("Caught");
        }
    }
}

```

## class Run

```

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

```

```

OneThread o1 = new OneThread();
TwoThread t1 = new TwoThread();
o1.start();
t1.start();

```

try

{  
Thread.sleep(5000);

catch(InterruptedException e){  
e.printStackTrace();

System.out.println("Interrupted");

o/p. BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

~~Output~~

Exception

class WrongAge extends Exception

{      static WrongAge (str)

}      public Super (str);

}

class Father

{

    int age;  
    Father (int age) throws WrongAge Exception

{

}      if (age < 0)

{

    throws new System.out.println ("Invalid age  
                              input");

    this. fAge = fAge;

{

public class Son extends Father

{

    int sonAge;

    public Son (int sonAge, int fAge) throws  
        WrongAge

{

        Super (fAge);

        if (sonAge > fAge)

{

            System.out.println ("Son's age can't be  
                              greater than father's age");

{

            this. sonAge = sonAge;

{

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            Father f = new Father(50);  
            Son s = new Son(70, 50);  
            catch (WrongAge e)  
        }  
    }  
}
```

2: Father's Age :- 60      Son's Age :- 41

~~8~~ ✓

Creating label, button and Text Field in a frame using AWT

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

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

```
public class AWTExample extends WindowAdapter
```

```
{
```

```
Frame f;
```

```
AWTExample()
```

```
{
```

```
f = new Frame();
```

```
f.addWindowListener(this);
```

```
Label l = new Label("Employee id:");
```

```
Button b = new Button("Submit");
```

```
TextField t = new TextField();
```

```
l.setBounds(20, 80, 80, 30);
```

```
t.setBounds(20, 100, 80, 30);
```

```
b.setBounds(100, 100, 80, 30);
```

```
f.add(b);
```

```
f.add(l);
```

```
f.add(t);
```

```
f.setSize(400, 300);
```

```
f.setTitle("Employee info");
```

```
f.setLayout(null);
```

```
f.setVisible(true);
```

```
}
```

```
public void windowClosing(WindowEvent e)
```

```
{
```

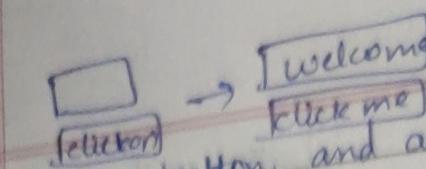
```
System.exit(0);
```

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

```
AWTExample awtObj = new AWTExample();
```

```
O/P
```

O/P



Create a button and add action listener for Mouse click.

```

import java.awt.*;
import java.awt.event.*;
public class EventHandling extends WindowAdapter
    implements ActionListener {
    Frame f;
    JTextField tf;
    EventHandling();
    {
        f = new Frame();
        f.addWindowListener(this);
        tf = new JTextField();
        tf.setBounds(60, 50, 170, 20);
        Button b = new Button("click me");
        b.setBounds(100, 120, 80, 30);
        b.addActionListener(this);
        f.add(b);
        f.add(tf);
        f.setSize(300, 300);
        f.setLayout(null);
        f.setVisible(true);
    }
}
  
```

```

f = new Frame();
f.addWindowListener(this);
tf = new JTextField();
tf.setBounds(60, 50, 170, 20);
Button b = new Button("click me");
b.setBounds(100, 120, 80, 30);
b.addActionListener(this);
  
```

```

f.add(b);
f.add(tf);
f.setSize(300, 300);
f.setLayout(null);
f.setVisible(true);
  
```

```

public void actionPerformed(ActionEvent e)
  
```

```

    tf.setText("Welcome");
  
```

```

public void windowClosing(WindowEvent e) {
    System.exit(0);
  
```

```

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

## Programs on IO

### Example 1

```
import java.io.*;  
public class ByteArrayInput {  
    public static void main (String [] args) throws  
        IOException {  
        byte [] buf = {35, 36, 37, 38};  
        ByteArrayInputStream b1 = new ByteArrayInputStream  
            (buf);  
  
        int k = 0;  
        while ((k = b1.read ()) != -1) {  
            char ch = (char) k;  
            System.out.println ("ASCII value of  
                character is : " + k + " special character is : " + ch);  
        }  
    }  
}
```

o/p ASCII value of character is 35: special character  
is : #

ASCII value of character is 36: Special character  
is : \$

ASCII value of character is 37: Special character  
is : %

ASCII value of character is 38: special character  
is : &

### Example 2

```
import java.io.*;  
public class ByteArray_ex {  
    public static void main (String args []) throws  
        Exception {  
        FileOutputStream fout1 = new FileOutputStream  
            ("Example1.txt");  
    }  
}
```

FileOutputStream fout2 = new FileOutputStream  
("Example2.txt");

Page

Byte Array Output Stream bout = new ByteArrayOutputStream();  
OutputStream();  
~~bout~~ bout.write(65);  
bout.writeTo(fout1);  
bout.writeTo(fout2);  
  
bout.flush();  
bout.close();  
System.out.println("Success...");

Output  
Success

Example 3:

public class FileEx

public static void main (String a []) throws  
IOException {  
fileInputStream fin = new FileInputStream  
("Example.txt");

byte [] bytes = new byte [20];

int i;

char c;

i = fin.read (bytes);

System.out.println ("Number of bytes read: " + i);

System.out.print ("Bytes read: ");

for (byte b: bytes) {

c = (char) b;

System.out.print (c);

}

g

Remaining Bytes: ]

Remaining Bytes: ]

that can be read: 0

dp

#### Example 4

```
import java.io.FileInputStream;
import java.io.IOException;
public class FileEx2
public static void main(String ar[]) throws
IOException {
    FileInputStream fin = new FileInputStream
("Example.txt");
    byte [] bytes = new byte [20];
    int i;
    char c;
    i = fin.read (bytes);
    System.out.println ("Num of bytes read: " + i);
    System.out.print ("Bytes read: ");
    for (byte b: bytes) {
        c = (char)b;
        System.out.print (c);
    }
}
```

Q.P

No. of bytes read: 20

Bytes read: Hello he is good.

S  
29/2/24