

I N D E X

Name Vishay Panwar N Std III sem Sec F

Roll No. 331 Subject Java School/College BMS

1. 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 Overload Demo {

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

```
Overload o=new Overload();
```

```
o.print(5);
```

```
o.print(7,13);
```

```
}
```

```
}
```

## 2. 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 z-dol, double z-pulser, double z-sugar) {
        total = z-dol * 100 + z-pulser * 50 + z - 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("Roma", "8060302010");
    Grocery g2 = new Grocery("Sharma", "7689652510");
    Grocery g3 = new Grocery("Bhama", "9636587412");
    g1.calc(2, 2, 1);
    g1.display();
    g2.calc(3, 5, 2);
    g2.display();
    g3.calc(1, 1, 0.5);
    g3.display();
}
```

## 3. import java.util.Scanner;

```
class QuadEq
```

```
int a, b, c;
double root1, root2, d;
```

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

```
void input()
```

```
{
```

```
System.out.println("Quadratic equation 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 solution. Roots are imaginary.");
    double realPart1 = (2 * a);
    double imaginary1 = Math.sqrt((d1 / (2 * a)));
    System.out.println("The equation has two complex roots " + real +
        " + " + imaginary1 + " i and " + real + " - " +
        imaginary1 + " i");
```

}

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

Million  
import java.util.Scanner;

```
class Student {
    String USN;
    String name;
    int marks[6] = new int[6];
    public void acceptDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your name: ");
        this.name = sc.nextLine();
        System.out.println("Enter your USN: ");
        this.USN = sc.nextLine();
        System.out.println("Enter marks for 6 subjects: ");
        for (int i = 0; i < marks.length; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            this.marks[i] = sc.nextInt();
        }
    }
```

3  
public double calculatePercentage()

```
    int totalMarks = 0;
    for (int mark : marks) {
        totalMarks += mark;
    }
    return (double) totalMarks / marks.length;
```

3

public void displayDetails()

```
    System.out.println("USN: " + this.USN);
    System.out.println("Name: " + this.name);
    System.out.println("Marks: ");
    for (int i = 0; i < marks.length; i++) {
        System.out.println("Subject " + (i + 1) + ": " + marks[i]);
```

3

```
system.out.println("Percentage" + calculatePercentage());
system.out.println();
```

3  
3

```
public class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter number of students:");
        int numStudents = sc.nextInt();
        Student [] students = new Student [numStudents];
        for (int i=0; i<numStudents; i++) {
            students[i] = new Student ();
            System.out.println("Enter details for student " +
                (i+1) + ":");
            students[i].acceptDetails ();
        }
        System.out.println("All student details:");
        for (Student student : students) {
            student.displayDetails ();
        }
    }
}
```

Output -

Enter number of students:

1

Enter details for student 1:

Enter your name:

Vishmay

Enter your UIN:

12

Enter marks for 6 subjects:  
subject 1: 34  
subject 2: 35  
subject 3: 36  
subject 4: 37  
subject 5: 38  
subject 6: 39

Student Details:

UIN: 12  
Name: Vishmay  
Marks:  
subject 1: 34  
subject 2: 35  
subject 3: 36  
subject 4: 37  
subject 5: 38  
subject 6: 39  
Percentage: 36.5%

2. import java.util.Scanner;

```

class Books
{
    String name;
    String author;
    int price;
    int numPages;

    Books() {
        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;
}

```

3

```

class Run
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in),
        int n;
        String name,
        String author;
        int price;
        int numPages;
    }
}

```

Soham 12/1/24

fallhort

### Q) Creating Abstract class shape

```
abstract class shape {
    int length;
    int width;
```

```
public shape(int length, int width) {
    this.length = length;
    this.width = width;
}

abstract void printArea();
```

```
}
```

```
class Rectangle extends shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }

    @Override
```

```
void printArea() {
    int area = length * width;
    System.out.println("Rectangle Area : " + area);
}
```

```
}
```

```
class Triangle extends shape {
    public Triangle(int length, int width) {
        super(length, width);
    }

    @Override
```

```
void printArea() {
    double area = 0.5 * length * width;
    System.out.println("Triangle Area : " + area);
}
```

```
}
```

class Circle extends shape {  
 public Circle (int radius) {  
 super (radius, 0);  
 }

}  
@Override

```
void printArea() {
    double area = Math.PI * (length * length);
    System.out.println("Circle Area : " + area);
}
```

}

public class Main {

```
public static void main (String [] args) {
    Rectangle rectangle = new Rectangle(5, 10);
    rectangle.printArea();
```

```
Triangle triangle = new Triangle(3, 6);
triangle.printArea();
```

```
Circle circle = new Circle(4);
circle.printArea();
```

}

Output -

Rectangle Area : 50

Triangle Area : 9.0

Circle Area : 50.26548245743669

2) import java.util.Scanner;

class Account {

    String customerName,  
    int accountNumber;  
    String accountType;  
    double balance;

    public 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 += amount;

        System.out.println("Deposit of \$ " + amount + " successful.  
                          Updated balance: \$" + balance);

}

    public void displayBalance() {

        System.out.println("Account balance: \$" + balance);

}

    public void depositInterest(double rate) {

        double interest = balance \* (rate / 100);

        balance += interest;

        System.out.println("Interest deposited: \$" + interest +  
                          " successful.  
                          Updated balance: \$" + balance);

}

Q) Java program to create class Account  
and implementing it

public void withdraw(double amount) {  
    if (balance >= amount) {  
        balance -= amount;  
        System.out.println("Withdrawal of \$" + amount + "  
                          successful. Updated balance: \$" +  
                          balance);  
    }  
    else {  
        System.out.println("Insufficient funds. Withdrawal failed.");  
    }  
}

Class CurAcc extends Account {

    double minimumBalance;  
    double serviceCharge;

    public CurAcc(String customerName, int accountNumber, double  
        minimumBalance, double serviceCharge) {

        super(customerName, accountNumber, "Current", balance);  
        this.minimumBalance = minimumBalance;  
        this.serviceCharge = serviceCharge;

}

@Override

    public void withdraw(double amount) {

        if (balance - amount >= minimumBalance) {  
            super.withdraw(amount);

        }  
    }

    else {

        System.out.println("Minimum balance not maintained.  
                          Service charge imposed. Updated  
                          balance: \$" + balance);

}

J

```
class Savings & extends Account {
```

```
    double interestRate;
```

```
    public Savings (String customerName, int accountNumber, double interestRate) {
```

```
        super (customerName, accountNumber, "Savings", 0.0);
```

this

```
    this.interestRate = interestRate;
```

```
    currentAccount.deposit(1000);
```

```
    currentAccount.displayBalance();
```

```
    currentAccount.withdraw(1500);
```

```
    currentAccount.checkMinimumBalance();
```

```
    currentAccount.close();
```

}

}

Output -

```
Deposit of ₹500 successful. Updated balance: ₹1500.0
```

```
Account Balance: ₹1500
```

```
Compound Interest deposited: ₹75 Updated balance: ₹1575.0
```

```
Withdrawal of ₹200.0 successful. Updated balance: ₹1375.0
```

```
Deposit of ₹1000.0 successful. Updated balance: ₹2375.0
```

```
Account Balance: ₹2375.0
```

```
Withdrawal of ₹1500 successful. Updated balance: ₹875.0
```

```
@Override
```

```
public void depositInterest(double rates) {
```

```
    double interest = balance * (rates / 100);
```

```
    balance += interest;
```

```
    System.out.println("compound interest deposited ₹" +  
    interest + ". Updated balance: ₹" +  
    balance);
```

}

}

```
public class Main {
```

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

```
        Scanner scanner = new Scanner (System.in);
```

```
        SavingsAccount savingsAccount = new Savings ("John Doe", 1234,  
        1000.0, 5.0);
```

```
        CurrentAccount currentAccount = new CurrentAccount ("Jane Smith",
```

```
        789012, 2000.0, 5.0,
```

```
        savingsAccount.deposit(500), 5.0);
```

```
        savingsAccount.displayBalance();
```

```
        savingsAccount.depositInterest(5), 5.0);
```

```
        savingsAccount.withdraw(200);
```

Explain  
19/11/24

## Packages

```
package CIE;
public class Student {
    String fName;
    int sem;
```

```
package CIE;
public class Internals extends Student {
    int[] internalMarks = new int[5];
```

```
package SEE;
import CIE.Student;
public class External extends Student {
    int[] externalMarks = new int[5];
```

```
import CIE.Internals;
import SEE.Externals;
public class Main {
    public static void main (String args) {
        Internals[] internalList = new Internals[10];
        Externals[] externalList = new Externals[10];
        for (int i=0; i<10; i++) {
            int[] finalMarks = new int[5];
            for (int j=0; j<5; j++) {
                finalMarks[j] = internalList[i].internalMarks[j] +
                    externalList[i].externalMarks[j];
            }
        }
    }
}
```

```
System.out.println("Final Marks for student " + (i+1) + ":");
for (int j=0; j<5; j++) {
    System.out.println("course " + (j+1) + " : " +
        finalMarks[j]);
```

Output -  
Final Marks for student 1

course 1: 50  
course 2: 60  
course 3: 70  
course 4: 40  
course 5: 85

Final marks for student 2:

course 1: 84  
course 2: 93  
course 3: 72  
course 4: 65  
course 5: 48

## Exception handling in Inheritance tree

```
class WrongAge extends Exception {
```

```
    WrongAge (String s)
```

```
    super(s);
```

```
}
```

```
}
```

```
class Father {
```

```
    int age;
```

```
    Father (int age) throws WrongAge {
```

```
        if (age < 0)
```

```
            throw new WrongAge ("Age cannot be negative");
```

```
        this.age = age;
```

```
}
```

```
}
```

```
class Son extends Father {
```

```
    int sonAge;
```

```
    Son (int fatherAge, int sonAge) throws WrongAge {
```

```
        super(fatherAge);
```

```
        if (sonAge >= fatherAge)
```

```
            throw new WrongAge ("Son's age should be less than  
                           Father's age");
```

```
        this.sonAge = sonAge;
```

```
}
```

```
}
```

```
public class ExceptionInheritance {
```

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

```
        try {
```

```
            Father father = new Father (50);
```

```
            Son son = new Son (50, 25);
```

```
            System.out.println ("Father's age: " + father.age);
```

```
            System.out.println ("Son's age: " + son.sonAge);
```

```
        } catch (WrongAge e) {
```

```
            System.out.println ("Exception caught: " + e.getMessage());
```

```
        }
```

3

3

object -

Father Age: 50  
Son's Age: 25

## Threads

```

class MessageThread extends Thread {
    private String message;
    private int interval;

    public MessageThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

## public class Thread Example :-

```

public static void main(String[] args) {
    MessageThread thread1 = new MessageThread("BMS
                                                college of Engineering, 10000);
    MessageThread thread2 = new MessageThread("CSE", 2000);
    thread1.start();
    thread2.start();
}

```

Output

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineers

CSE

CSE

CSE

CSE

CSE

18/2/24

20/11/2017

1. creating label, button and Textfield 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();
        t.setBounds(20, 20, 80, 30);
        b.setBounds(20, 100, 80, 30);
        t.setBounds(100, 100, 80, 30);
        f.add(l);
        f.add(t);
        f.add(b);
        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 awt_obj = new AWTExample();
    }
}

```

Output -

Employee info  
Employee Id :  [Submit]

2. Create a button and add a actionlistener for Mouse Click.

```

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

class EventHandling extends WindowAdapter implements
    ActionListener {
    Frame f;
    TextField tf;

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

}

Output

[ ]

Value

[which we]

### 3. Example :

```
import java.io.*;  
public class ByteArrayInput {  
    public static void main (String [ ] args) throws IOException {  
        byte [ ] buf = { 25, 36, 37, 38 };  
        ByteInputStream byt = new ByteInputStream (buf);  
        int k = 0;  
        while ((k = byt.read ()) != -1) {  
            char ch = (char) k;  
            System.out.println ("ASCII value of character  
                is : " + k + "; special character is : " + ch);  
        }  
    }  
}
```

Output -

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 : &

### 4. Example :-

import java.io.\*;

```
public class ByteArrEx {  
    public static void main (String args[]) throws IOException {  
        FileOutputStream fout1 = new FileOutputStream ("Example.txt");  
        FileOutputStream fout2 = new FileOutputStream ("Example.txt");  
        byte [] outputstream = new ByteOutputStream ();  
        bout.write (05);  
        bout.writeTo (fout1);  
        bout.writeTo (fout2);  
        bout.flush ();  
        fout1.close ();  
        System.out.println ("Success... ");  
    }  
}
```

Output :

Success...

### 5. Example 3 :

```
public class FileEx {  
    public static void main (String args[]) throws IOException {  
        FileInputStream fin = new FileInputStream ("Example.txt");  
        int content;  
        System.out.println ("Remaining bytes that can be read : "  
            + fin.available());  
        content = fin.read ();  
        System.out.print ((char) content + " ");  
        System.out.print (content + " ");  
        System.out.println ("Remaining bytes that can be read : "  
            + fin.available());  
        System.out.println ("Remaining bytes that can be read : "  
            + fin.available());  
    }  
}
```

output

remaining bytes that can be read : ,

2

65

remaining bytes that can be read : 0

### Ex. Example 54

```
import java.io.FileInputStream;
```

```
import java.io.IOException;
```

```
public class FileEx2 {
```

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

```
        FileInputStream fin = new FileInputStream("file.txt");
```

```
        byte b[] 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);
```

```
}
```

```
}
```

```
}
```

Output:

Number of bytes read : 20

Bytes read : Hello in vi now

good

Sree  
23/2/24