

- 1) Write a program to overload the method to print that prints sum of n natural numbers when one variable is passed and prints the prime numbers in a given range when 2 parameters are passed.

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")

```
if ("+sum);
```

```
}
```

```
void print (int m, int n) {
```

System.out.println ("Prime numbers in the range

```
one");
```

```
for (int i=m; i<n; i++) {
```

```
    int flag = 0;
```

```
    for (int j=2; j<=i; j++) {
```

```
        if (i % j == 0) {
```

```
            flag = 1;
```

```
        break;
```

```
}
```

```
}
```

```
if (flag == 0)
```

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

```
}
```

```
3
```

class OverloadDemo {

public static void main(String[] args) {

Overload o = new Overload();

o.print(5);

o.print(7, 13);

3.

3.

② Output: Sum of 10 natural numbers is 55

Prime numbers in the range are

5

7

11

13

19.

2) Write a Java program to create a class Grocery that has variables c-name and c-phone. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses and quantity of sugar. The method to return the total price. Display the name, ph-no and total bill of 3 customers.

class OttGrocery {

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;

3

void calc(double q-dal, double q-pulses, double q-sugar)
{

 total = q-dal * 100 + q-pulses * 80 + sugar * 50;
}

void display()

 System.out.println("Name " + " " + "Phone number
 " + " " + "Total");

 System.out.println(c-name + " " + c-ph + " " + total);

 System.out.println();

}

3

class Demo {

 public static void main (String[] args) {

 Grocery g1 = new Grocery ("Rama", "8060302010");

 Grocery g2 = new Grocery ("Shamo", "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();

3

3.

Output:

Name	phone number	Total
Rama	8618372865	410.0

Name	phone number	Total
Viraj	9545721582	800

Name	phone number	Total
Sid.	9485672838	205

~~Write a java program~~

88
12
12

1. Write a Java program to create a class student with members USN, name, marks (6 subjects). Include methods to accept student details and marks. Also include a method to calculate the percentage and display appropriate details. (Array of students object to be created).

import java.util.Scanner;

class Student {

String USN;

String name;

int[] marks = new int[6];

void acceptDetails() {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter USN:");

USN = scanner.next();

System.out.println("Enter Name:");

name = scanner.next();

System.out.println("Enter marks from 6 subjects:");

for (int i=0; i<6; i++) {

System.out.println("Subject " + (i+1) + ":");

marks[i] = scanner.nextInt();

}

3.

double calculatePercentage() {

int totalMarks = 0;

for (int mark : marks) {

totalMarks += mark;

4.

return (double) totalMarks / 6;

5.

void displayDetails() {

System.out.println ("Student details:");

System.out.println ("USN: " + USN);

System.out.println ("Name: " + Name);

System.out.println ("Percentage: " + calculatePercentage
+ "%");

}

}

public class exp {

public static void main (String args) {

Scanner scanner = new Scanner (System.in);

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

int numStudents = scanner.nextInt();

student [] students = new student [numStudents];

for (i=0; i < numStudents; i++) {

student [i] = new student();

System.out.println ("In Enter details for
student " + (i+1) + ":");

student [i].acceptDetails();

System.out.println ("In Details of all students:");

for (student student : students) {

student.displayDetails();

System.out.println ();

}

}

}

Output:

Enter the number of students: 9

Enter details for student 1:

Enter USN: 2023BMS01

Enter name: Syed.

Enter marks for 6 subjects:

Subject 1: 80

Subject 2: 90

Subject 3: 90

Subject 4: 90

Subject 5: 90

Subject 6: 90

Enter details for student 2:

Enter USN: 2023BMS02

Enter name: Farhan.

Enter marks for 6 subjects:

Subject 1: 90

Subject 2: 90

Subject 3: 90

Subject 4: 90

Subject 5: 90

Subject 6: 90

~~Details of all students:~~

~~Student details:~~

~~USN: 2023BMS01~~

Name: Syed.

Percentage: 83.33%

~~Student details:~~

~~USN: 2023BMS02~~

Name: Farhan

Percentage: 90.0%

2) Create a class Book that contains four members: name, author, price, and numPages. Include a constructor to set the value for the members. Include methods to set and get the details of the object. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

Import java.util.Scanner;

class Book {

String name;

String author;

int price;

int numPages;

Book() { }

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;

}

2.

class Main

{

public static void main (String args[]){

Scanner s = new Scanner (System.in);

int n;

String name;

String author;

int price;

int numpage;

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

n = s.nextInt();

Book b[];

b = new Book[n];

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

{

System.out.println ("Book" + (i+1) + ":");

System.out.println ("Enter name of Book: ");

name = s.next();

System.out.println ("Enter author: ");

author = s.next();

System.out.println ("Enter price: ");

price = s.nextInt();

System.out.println ("Enter no. of page: ");

numpage = s.nextInt();

b[i] = new Book (name, author, price, numpage);

3.

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

System.out.println ("Book" + (i+1) + "In" + b[i]);

3.

?

Output

Enter the number of book : 2

Book 1 :

Enter the name of the Book : Jungle-Book

Enter the ^{author} name of the Book : Rudyard.

Enter the price of the Book : 1000

Enter the number of page of the Book : 500

Book 2 :

Enter the name of the Book : Tales of Alkon.

Enter the author of the Book : Bibbal.

Enter the price of the Book : 900

Enter the number of the page of the Book : 400

Book 1 :

Book name : Jungle - Book.

Author : Rudyard

Price : 1000

Number of page : 500

Book 2 :

Book name : Tales - of - Alkon

Author : Bibbal.

Price : 900

Number of page : 400

CSO
12/10/17

1. Develop a Java program to create an abstract class named shape that contains two integers and an empty method named printArea(). provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the shape class. Each one of the classes contain the method printArea() that prints the area of the given shape.

abstract class shape {

protected int length;

protected int width;

public shape (int length, int width) {

this.length = length;

this.width = width;

3.

public abstract void printArea();

3.

class Rectangle extends shape {

public Rectangle (int length, int width) {

super (length, width);

3.

public void printArea() {

int area = length * width;

System.out.println ("Rectangle Area: " + area);

3.

3.

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

public 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);
 }
}

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

public void printArea()
double area = Math

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

Rectangle rectangle = new Rectangle (5, 10);
 rectangle.printArea();

Triangle triangle = new Triangle (3, 8);

triangle.printArea();

Circle circle = new Circle (4);

circle.printArea();

3.

3.

Output:

Rectangle Area: 50

Area of Triangle: 17

Area of Circle: 50.

- 2) Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the class current and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following steps.

- a) Accept deposit from customer and update the balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance.

check for the minimum balance, impose penalty if necessary and update the balance.

import java.util.Scanner;

class Account {

protected String customerName;

protected long accountNumber;

protected String accountType;

protected double balance;

public Account (String customerName, long accountNumber, String accountType, double balance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance;

}

public void displayBalance() {

System.out.println ("Account Number: " + accountNumber);

System.out.println ("Customer Name: " + customerName);

System.out.println ("Account Type: " + accountType);

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

}

public void deposit (double amount) {

balance += amount;

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

displayBalance();

3.

public void withdraw (double amount) {

if (amount <= balance) {

balance -= amount;

System.out.println ("Withdrawal of \$" + amount)

"successful");

else {

System.out.println("Insufficient funds. Withdrawal failed.");

}

displayBalance();

}

.

close current account. Account class.

private double minimumBalance = 1000;

private double serviceCharge = 50;

public CurrentAccount(String customerName, long accountNumber, double balance) {

super(customerName, accountNumber, "current", balance);

public void withdraw(double amount) {

if (amount <= balance - minimumBalance) {

balance -= amount;

System.out.println("Withdrawal of \$" + amount + " successful");

else {

System.out.println("Insufficient funds. Withdrawal failed. Service charge of \$" + serviceCharge + " imposed");

balance -= serviceCharge;

}

displayBalance();

}

class SavAcct extends Account {

private double interestRate = 0.05;

public SavAcct(String customerName, long accountNumber, double balance) {
Super(customerName, accountName, "savings", balance);

System.out.println("Interest rate is " + interestRate);
balance += interestRate * balance;
System.out.println("New balance is " + balance);

3.

public void computeInterest() {

double interest = balance * interestRate;

balance += interest;

System.out.println("Interest of \$" + interest + "
credited);

displayBalance();

4.

public class exp2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

SavAcct savingsAccount = new SavAcct("Fashion", 123456789, 5000);

savingsAccount.displayBalance();

savingsAccount.deposit(1000);

savingsAccount.computeInterest();

savingsAccount.withdraw(2000);

Current Account currentAccount = new CurrentAcc("Fashion", 987654321, 1500);

currentAccount.displayBalance();

currentAccount.deposit(500);

Current Account withdraw (\$2000);

Scanner closed;

3

3

Output:

Account Number: 123456789.

Customer Name: Farhan.

Account Type: savings.

Balance: \$ 5000.0.

Deposit of \$1000.0 successful.

Account Number: 123456789.

Customer Name: Farhan

Account Type: savings.

Balance: \$ 6000.0.

Interest of \$300.0 credited.

Account Number: 123456789

Customer Name: Farhan

Account Type: savings.

Balance: \$ 6300.00.

withdrawal of \$2000.0 successful.

.

✓
Ans
19.01.21

Week - 5

1) CIE package

```
package CIE;
public class Personel {
    public String usn;
    public String name;
    public int sem;
    public Personel (String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

Import java.util.ArrayList;

```
public class Internode {
    public List<InternalMarks> internalMarks;
    public Internode (List<InternalMarks> internalMarks) {
        this.internalMarks = internalMarks;
    }
}
```

package SEE;

Import CIE.Personel;

```
public class External extends Personel {
    public List<SeeMarks> seeMarks;
    public External (String num, String name, int sem,
                    List<SeeMarks> seeMarks)
}
```

Super (usn, name, sem);

~~this.SeeMarks = SeeMarks;~~

g

}

```
package FormMain;  
import java.util.ArrayList;  
import CIF.Internal;  
import CIF.Personal;  
import SEE.External;  
public class Main {  
    public static void main (String [] args) {  
        int n=3;  
        Student[] students = new Student [n];  
        for (int i=0; i<n; i++) {  
            int [] InternalMarks = { 80, 75, 90, 85, 95 };  
            int [] SeeMarks = { 30, 80, 75, 90, 85 };  
            students [i] = new Student (new Personal ("USN"+i,  
                "Student"+i), new Internal (InternalMarks));  
            students [i].See = new External ("USN"+i, "Student"+i, 1,  
                SeeMarks);  
        }  
        for (int i=0; i < students.length; i++) {  
            Student student = students [i];  
            System.out.println ("Student: "+ student.Personal.name);  
            System.out.println ("Internal Marks: "+ ArrayList.toString (student.Internal.  
                InternalMarks));  
            System.out.println ("SEE Marks: "+ ArrayList.toString (student.See.SeeMarks));  
        }  
    }  
}
```

static class Student.

```
public Personal personal;  
public Internal internal;  
public External see;  
public Student (personal personal, internal internal) {  
    this.personal = personal;  
    this.internal = internal;  
}
```

output:

Student . Student 0

Internal Marks : [80, 75, 90, 85, 95]

SFF Marks : [70, 80, 75, 90, 85]

Student . Student 1

Internal Marks : [80, 75, 90, 85, 95]

SFF Marks : [70, 80, 75, 90, 85]

Student . Student 2

Internal Marks : [80, 75, 90, 85, 95]

SFF Marks : [70, 80, 75, 90, 85].

Week - 6.

- 3) Write a program that demonstrates handling of exception.
 In inheritance derive. Create base class called Father and derived class called Son which extends the base class.
 In Father class, Implement a constructor which takes the age and throwing the exception WrongAge() , when the input age < 0 . In son class, Implement a constructor that takes both father and Son's age and throwing an exception if son's age is \geq father's age.

class Father {

public int age;

Father (int age) {

if (age >= 0) {

throw new IllegalArgumentExpection ("Age cannot be -ve");

}

this.age = age;

}

3.

public class Son extends Father {

public int SonAge;

public Son (int fatherAge, int SonAge) {

Super (fatherAge);

if (SonAge \geq fatherAge) {throw new IllegalArgumentExpection ("son age cannot be
 \geq father age");

}

7

this.SonAge = SonAge;

3

3-

```
import java.util.Scanner;  
public class Main {  
    public static void main (String[] args) {  
        Scanner s = new Scanner (System.in);  
        try {  
            System.out.print ("Enter father age: ");  
            int fatherAge = s.nextInt ();  
            System.out.print ("Enter son's age: ");  
            int sonAge = s.nextInt ();  
            Son son = new Son (fatherAge, sonAge);  
            System.out.println ("Father age: " + son.age);  
            System.out.println ("Son's age: " + son.sonAge);  
        }  
    }  
}
```

```
catch (IllegalArgumentException e) {  
    System.out.println ("Exception: " + e.getMessage());  
}
```

c.close();

3.

Output:

Enter Father's age: 20

Enter Son's age: 40

Exception: Son's age cannot be \geq father's age.

- 3) Write a program which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds and another displaying "CSF" once every two seconds.

```
class BMSDisplay extends Thread {
    public void run() {
        while (true) {
            System.out.println ("BMS college of Engineering");
            Thread.sleep (10000);
        }
    }
}
```

```
class CSFDisplay extends Thread {
    public void run() {
        while (true) {
            System.out.println ("CSF");
            Thread.sleep (2000);
        }
    }
}
```

3 catch (InterruptedException e).

e.printStackTrace();

}

}

}

public class Main {

 public static void main (String [] args) {

 BMSDisplay bmsThread = new BMSDisplay();

 CSEDipplay cseThread = new CSEDipplay();

 bmsThread.start();

 cseThread.start();

}

,

Output:

BMS College of Engineering.

CSE

CSE

BMS College of Engineering.

CSE

CSE

CSE

~~CDL
R3
16.02.2023~~

Output

Employee Id:

2023A Submit

1. Create 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();

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 awt obj = new AWTExample ();

}

,

2. Create a button and add a action listener for Mouse click

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

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

```
public 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(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();
```

}

}

Output:

HP

Welcome

click me

click me

Programme on IO

1. Import java.io.*;

public class ByteArrayInput {

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

byte [] buf = { 35, 36, 37, 38 };

ByteArrayInputStream byt = new ByteArrayInputStream
(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);

}

2

3.

Guru

2. Import java.io.*;

public class ByteArrayInput {

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

byte [] buf = {35, 36, 37, 38};

ByteArrayInputStream byt = new ByteArrayInputStream
(buf);

int k=0;

while (k = byt.read () != -1) {

char ch = (char) k;

System.out.println ("ASCII value of character is:
" + k + ": Special character ie: " + ch);

}

}

3. public class FileEx {

public static void main (String args) throws
IOException {

FileInputStream fin = new FileInputStream

("Example.dat");

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());

3

3.

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

5) `import java.io.*;`
`public class ByteArrayEx2`
`public static void main (String args) throws`
`Exception {`
~~`FileOutputStream fout1 = new FileOutputStream`
`("Example.txt");`
`FileOutputStream fout2 = new FileOutputStream`
`("Example.txt");`
`ByteArrayOutputStream bout = new ByteArrayOutputStream();`
`bout.write (bs);`
`bout.writeTo (fout1);`
`bout.writeTo (fout2);`~~

bout. filypb(2);

bout. ridge(2);

System.out.println("Swallow...");

3.

3

~~Q & B~~ 0.1 .01
~~Q & B~~ 0.2 .02
~~Q & B~~ 0.3 .03