

INDEX

Name : Venkatesh V. Chandru Class : 3-nd sum

Section : 3F Roll No. : Subject : P.O.O.T

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

Code:

```
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 nos 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;
        }
    }
}
```

```
class OverloadDemo {
    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

- 2 Write a Java program to create a class Grocery that has the 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.

Code

```
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 " + " " + "Name " + " " + "  
    Phone number " + " " + "Total ");  
}
```

System.out.println(cname + " " + cph + " " + total),

System.out.println(),

y

3

class Cdemo1

public static void main(String[] args) {

Grocery g1 = new Grocery("Ruma", "12345"),

Grocery g2 = new Grocery("Shama", "23456"),

Grocery g3 = new Grocery("Bhima", "67891");

g1.calc(2, 2, 1),

g1.display(),

g2.calc(3, 5, 2);

g2.display()

g3.calc(1, 1, 0.5)

g3.display()

y

j

3

Output:

Name	Phone number	Total
Ruma	12345	
Name	Phone number	Total
Shama	23456	800 0
Name	Phone number	Total
Bhima	67891	2050

3. Write program for 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("Enter a in form ax²+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()  
{
```

```
    if d = (b*b) - (4*a*c);  
    }
```

```
    void calculateRoots()  
{
```

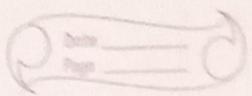
```
    if (d > 0)  
    { System.out.println("Roots are real and unequal");  
      root1 = (-b + Math.sqrt(d)) / (2*a);  
      root2 = (-b - Math.sqrt(d)) / (2*a);  
    }
```

```
    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 if ($a < 0$)

```
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 eqn has 2 complex roots: " + real + " + " +  
"imaginary * i" + " and " + real + " - " + imaginary + " i");
```

3

4

5

Class Main {

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

```
{ Quad q = new Quad();
```

```
q.setd0();
```

```
q.discriminant ();
```

```
q.calculateRoots ();
```

3

4

Date: 12/01/2024

1 Write a java program to create a class Student with USN, name, marks(6 subjects). Include methods to accept details and marks. Also include a method to calculate the % and display the appropriate details.

Input:

```
import java.util.Scanner;  
public class Main {  
    public static void Main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        Student[] students = new Student[5];  
        for (int i = 0; i < students.length; i++) {  
            students[i] = new Student();  
            System.out.println("enter details of student " + (i + 1));  
            students[i].acceptDetails(sc);  
            students[i].acceptMarks(sc);  
        }  
        for (int i = 0; i < students.length; i++) {  
            students[i].calc();  
            students[i].disp();  
        }  
        sc.close();  
    }  
}
```

```
class Student {  
    private String usn;  
    private String name;  
    private int[] marks = new int[6];  
    private int totalMarks;  
    private double percentage;  
    public void acceptDetails(Scanner sc) {  
        System.out.print("enter usn: ");  
        usn = sc.nextLine();  
        System.out.print("enter name: ");  
        name = sc.nextLine();  
    }
```

```

public void acceptMarks(Scanner sc) {
    for(int i=0; i<marks.length, i++) {
        System.out.print("enter marks for subject " + (i+1) + ": ");
        marks[i] = sc.nextInt();
    }
    sc.nextLine(); // 3
}

public void calculate() {
    totalmark = 0;
    for(int mark : marks) {
        totalMark += mark; // 3
    }
    percentage = (totalmark / 600.0 * 100); // 3
}

public void disp() {
    System.out.println("\nStudent details: ");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Total marks " + totalmark);
    System.out.println("Percentage: " + percentage + "%"); // 3
}

```

Output

enter details of student 1 enter usn: 123

enter name ram enter

enter marks for subject 1: 30

enter marks for subject 2: 31

enter marks for subject 3: 34

enter marks for subject 4: 37

enter marks for subject 5: 39

enter marks for subject 6: 31

Student details.

usn: 123

name: ram

Total mark marks: 202

Percentage: 33.66%

2. ① Create a class Book that contains 4 members: name, author, price and num pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Input:

```
import java.util.Scanner  
  
public class Main {  
  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter no of books: ");  
        int n = sc.nextInt();  
        Book[] books = new Book[n];  
  
        for(int i = 0; i < n; i++) {  
            System.out.println("Enter details for Book " + (i+1) + ":");  
            System.out.print("Name: ");  
            String name = sc.next();  
            System.out.print("Author: ");  
            String author = sc.next();  
            System.out.print("Price: ");  
            System.out.printf("double price = sc.nextDouble();  
            System.out.print("Number of pages: ");  
            int numPages = sc.nextInt();  
            books[i] = new Book(name, author, price, numPages);  
            System.out.println(books[i]);  
        }  
        sc.close();  
    }  
}
```

class Book {

String name;

String author;

double price;

int numPages;

public Book(String name, String author, double price, int numPages) {

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages; }

public String getName() {

return name; }

public void setName(String name) {

this.name = name; }

public String getAuthor() {

this.author = author; }

public void setAuthor(String author) {

this.author = author;

public double getPrice() {

return price; }

public void setPrice(double price) {

this.price = price; }

public intgetNumPages() {

return numPages; }

public void setNumPages(int numPages) {

this.numPages = numPages; }

public String toString() {

return "Book [name=" + name + ", author=" + author + ", price = " + price + ", numPages=" + numPages + "]"; }

~~Enter~~ Enter the number of books:

1

Enter details for book 1:

Name: ram

Author: sham

Price: 123

Number of pages: 12

Book [name = ram, author = sham, price = 123.0, num. pages = 12]

1. Develop a java program to create an abstract class named `Shape` that contains 2 integers and an empty method named `printArea()`. Provide 3 classes named `rectangle`, `triangle` and `circle` such that each one of these classes extends the class `shape`. Each one of these classes contains the method `printArea()` that prints the area of the given shape.

Input:

```

abstract class Shape {
    int length;
    int width;
    Shape(int length, int width) {
        this.length = length;
        this.width = width;
    }
    abstract void printArea();
}

class rectangle extends Shape {
    rectangle(int length, int width) {
        super(length, width);
    }
    void printArea() {
        System.out.println("Area: " + (length * width));
    }
}

class triangle extends Shape {
    triangle(int length, int width) {
        super(length, width);
    }
    void printArea() {
        System.out.println("Area: " + (0.5 * length * width));
    }
}

class circle extends Shape {
    circle(int length, int width) {
        super(length, width);
    }
    void printArea() {
        System.out.println("Area: " + (3.14 * length * width));
    }
}

```

2
class MainOf

{public static void main(String args[]) {
rectangle a = new rectangle(3, 4);

a.printArea();

triangle b = new triangle(3, 4);
b.printArea();

circle c = new circle(3, 3);

c.printArea();

3 3

Output

Area: 12

Area: 6.0

Area: 28.26

7. Develop a java program to create a class Bank that maintains 2 kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and ~~allows~~ withdrawal facility but no cheque book facility. The current account provides check book facility but no interest. Current account holders should also maintain a min. balance & 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 classes current and savings to make them more specific to their requirements. Each includes the methods in order to achieve the following tasks:

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

Check for the min. baln., impose penalty if necessary & update the balance.

Input:

class Account {

String String cname;

int accno;

String acctype;

double balance;

Account (String ^a cname, int ^b accno, a String ^c acctype, double ^d balance) {

this cname = ^a cname;

this accno = ^b accno;

this acctype = ^c acctype;

this balance = ^d balance; }

29

~~class Current extends Account { double minbalance; minbalance, minbalance~~

~~current(String a, int b, String c, double d) {~~

~~super(a, b, c, d);~~

~~this.minbalance = 1000;~~

~~this.mservicecharge = 50; }~~

~~public void withdraw(double amount) {~~

~~if (balance - amount >= minbalance) {~~

~~balance -= amount;~~

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

~~} else {~~

~~System.out.println("Funds less service charge imposed.");~~

~~balance -= mservicecharge; }~~

~~}~~

~~System.out.println("Current balance: " + balance); }~~

~~class Account {~~

~~String chname;~~

~~int accno;~~

~~String acctype;~~

~~double bal;~~

~~Account(String a, int b, String c, double d) {~~

~~this.chname = a;~~

~~this.accno = b;~~

~~this.acctype = c;~~

~~this.bal = d;~~

~~}~~

~~public void disp() {~~

~~System.out.println("balance: " + bal); }~~

~~public void deposit(double am) {~~

~~balance += am; }~~

~~public void withdraw(double am) {~~

~~if (bal >= am) {~~

~~bal -= am; }~~

~~System.out.println(" " + am + " withdrawn"); }~~

```

class curracc extends Account {
    double minbal;
    double servch;
    curracc (String a, int b, String c, double d) {
        super(a, b, c, d);
        this.minbal = 1000;
        this.servch = 50; Wd
    }
    public void Wd(double am) {
        if (balance - am >= minbal) withdraw(am); q
        else {
            balance -= servch;
            System.out.println("Service charge imposed."); q
        }
    }
}

```

Surender
19/11/20

```

class savacc extends Account {
    double interestrate;
    double interest;
    Account savacc (String a, int b, String c, double d) {
        super(a, b, c, d);
        this.interestrate = 0.05;
        this.interest = 0.05 * d; q
    }
    public void comp() {
        bal = bal + interest; q q
    }
    public void wd(double am) {
        if (bal - am >= minbal) withdraw(am); q
        else printf("withdrawal not possible."); q q
    }
}

```

```

class main {
    public static void main(String args[]) {
        savacc a = new savacc ("ram", 23, "xyz", 10.5);
        curracc b = new curracc ("john", 345, "abc", 12.5);
        a.comp();
        a.wd(20); a.withdraw(20);
    }
}

```

b. wd(50);

a. disp();

b. disp();

yy

O/P:

20 withdrawn

50 withdrawn

balance :

~~50~~
~~25~~ 11/24

16/12/2024

Date _____
Page _____

1 Package CIE

```
-> package CIE;  
import java.util.Scanner;  
public class Student {  
protected String urn = new String();  
protected String name = new String();  
protected int sum;  
  
public void inputStudentDetails(){  
Scanner sc = new Scanner(System.in);  
System.out.println("give urn: ");  
urn = sc.nextLine();  
System.out.println("give name: ");  
name = sc.nextLine();  
System.out.println("give sum: ");  
sum = sc.nextInt();  
}
```

```
package CIE;  
import java.util.Scanner;  
protected int marks[] = new int[5];  
public void inputCIEMarks() {  
Scanner sc = new Scanner(System.in);  
for(int i=0; i<5; i++) {  
System.out.println("Enter marks for course " + (i+1));  
marks[i] = sc.nextInt();  
}
```

3

```
package SEE;  
import CIE.internals;  
import java.util.Scanner;  
public class Editorial extends Internals {
```

protected int marks[];
protected int finalmarks[],

3

class External() {

marks = new int[5];

finalmarks = new int[5].

4

public void info() {
SEE members();

Scanner sc = new Scanner(System.in);

for (int i=0; i<5; i++) {

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

marks[i] = sc.nextInt();

public void calcFinalMarks() {

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

finalMarks[i] = marks[i]/2 + rubMarks[i];

5

backing SEE;

public class External extends Internal() {

External();

public class Main {

public static void main(String args[]) {

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

finalMarks[i] = new External();

finalMarks[i] = new External();

finalMarks[i] = new CIEMark();

6

System.out.println("Display Data: ");

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

finalMarks[i].calcFinalMarks();

7

2. Exception Handling

-> class Father {

 public int age;

 Father (int age) {

 if (age < 0) {

 throw new IllegalArgumentException ("age cannot be negative");

 }

 this.age = age;

}

 }

 public class Son extends Father {

 public int SonAge;

 public Son (int fatherAge, int sonAge) {

 super (fatherAge);

 if (sonAge >= fatherAge) {

 throw new IllegalArgumentException ("Son's age cannot be greater than father's age");

 }

 this.SonAge = sonAge;

}

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

Output:

enter father's age: 20

enter son's age: 40

Exception: Son's age cannot be greater than father's age.

3. Threads

class abc extends Thread {

String a;

int i;

abc(String a, int i) {

this.a = a;

this.i = i;

}

public void run() {

try {

while (true) {

System.out.println(a);

Thread.sleep(1);

}

catch (InterruptedException e) {

e.printStackTrace();

}

4

class abc {

public static void main(String args[]) {

abc t1 = new abc("BMS", 10000);

abc +2 : now abc ("CSE", 20000);

+1 start();

+2 start();

y

y

Output:

BMS

CSE

CSE

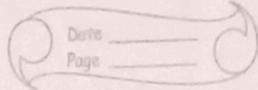
CSE

CSE

CSE

BMS

Date: 23/02/2024



1 Create label , button & 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);

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

f.add(l);

f.add(t);

f.add(b);

f.setSize(400, 300);

f.setTitle("Employee info");

b.setLayout(null);

f.setVisible(true);

2

public void windowClosing(WindowEvent e) {

System.exit(0);

3

public void main(String[] args) {

AWTExample awtObj = new AWTExample();

4

5

Output:

Employee Info

Employee id

1234

Submit

2 Create a button & add an action listener for mouse click.

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

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

public class EventHANDLER extends WindowAdapter implements ActionListener {

Frame f;

private JTextField tf;

EventHANDLER() {

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 EventHANDLER();

y

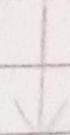
y

Output:

abc

click me

button
clicked



Welcome

click me

Programs on IO

1 Example 1

import java.io.*;

public class ByteArrayInput {

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

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

ByteArrayInputStream bf = new ByteArrayInputStream(buf);

int k = 0;

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

char ch = (char) k;

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

}

y

z

2 Example 2 :

import java.io.*;

public class ByteArrayEx2

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

FileInputStream fint1 = new FileInputStream("Example1.txt");

FileOutputStream fint2 = new FileOutputStream("Example2.txt");

ByteArrayOutputStream bout = new ByteArrayOutputStream();

bout.write(65);

bout.writeTo(fint1);

bout.writeTo(fint2);

bout.flush();

bout.close();

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

y y

3.

Example 3

```
public class FileEx3 {
```

```
    public static void main(String ar[]) 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());
    }
```

3

4.

Example 4

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

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

```
public class FileEx4 {
```

```
    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("Number of bytes read: " + i);
```

```
        System.out.println("Bytes read: ");
```

for (byte b : bytes) {

c = (char) b;

System.out.print(c);

}

y

y

Output :

Ex. 1

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 : &

Ex. 2 :

SUCCESS . . .

Ex. 3 :

Hello World !

Remaining bytes that can be read : 4

\$H E L L O W O R L D !

Remaining bytes that can be read : 0

Ex. 4 :

Number of bytes read : 1

bytes read : A

Prache
23/2/24