

22/02/2023

- 1) Write a program to overload the method 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 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();

0. print(5);

0. print(7,13);

3

3

Output Sum of 5 natural numbers is 15

prime numbers in the range are

7

11

13

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

=> 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-pulse, 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();  
}  
}
```

3

class GDemo{

```
public static void main (String [] args){  
Grocery g1 = new grocery ("Rama", "8060302010");  
Grocery g2 = new grocery ("Shoma", "9481520238");  
Grocery g3 = new grocery ("Bhama", "9448722133");  
g1.calc(2,2,1);  
g1.display();  
g2.calc(3,5,2);  
g2.display();  
g3.calc(1,1,0.5);  
g3.display();
```

}

3

Output: ~~Name phone number Total~~

Name	phone number	Total
Rama	8060302010	410

~~Name phone number Total~~

Name	phone number	Total
Shoma	9481520238	800

~~Name phone number Total~~

Name	phone number	Total
Bhama	9448722133	205.

Q3. Write a Java program to calculate roots of quadratic equation
use appropriate methods to take inputs and calculate the root

→ 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 : " + a + "x² + 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.print ("Roots are equal");
```

```
    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.print ("Roots are real and equal");
```

```
    root1 = (-b + Math.sqrt(d)) / (2*a);
```

```
    System.out.println ("Roots: " + 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 ();
```

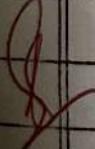
```
        q.input();
```

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

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

```
}
```

```
}
```



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

Enter a : 5

Enter b : 10

Enter c : 30

No real solutions. Roots are imaginary

The equation has two complex roots : $-1.0 + 2.23606797i$ and $-1.0 - 2.23606797i$

12/01/2024

LAB -

Q. 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 student object to be created).

```
→ import java.util.Scanner;
class student {
    String usn;
    String name;
    int marks[] = new int[6];
    void Details() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN");
        usn = s.nextLine();
        System.out.println("Enter name");
        name = s.next();
        System.out.println("Enter marks for 6 subjects");
        for (int i = 0; i < 6; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }
    double percentage() {
        int total = 0;
        for (int i = 0; i < 6; i++) {
            total += marks[i];
        }
        double percentage = (total / 600) * 100;
        return percentage;
    }
}
```

`total += marks[i];`

`double p = total / 6;`

`return p;`

`}`

`void play()`

`{`

`System.out.println("In Student Details:")`

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

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

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

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

`{`

`System.out.println("Subject " + (i+1) + ":" + marks[i]);`

`}`

`System.out.println("percentage: " + percentage() + "%");`

`}`

~~`class Lab1 Student`~~

`{`

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

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

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

`int n = s.nextInt();`

`Student[] Students = new Student[n];`

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

`{`

`Student[i] = new Student();`

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

`Student[i].details();`

`}`

for (Student student : students)
{
 student.display();
}

Output

Enter the number of Students : 1

Enter the details of Students 1

Enter USN

IBM22CS200

Enter name

Ravi 078

Enter marks for 6 subjects:

Subject 1 : 90

Subject 2 : 99

Subject 3 : 98

Subject 4 : 89

Subject 5 : 89

Subject 6 : 89

Student details

USN: IBM22CS200

Name: Ravi

Subject 1 : 90

Subject 2 : 99

Subject 3 : 98

Percentage : 92.0%

Subject 4 : 89

Subject 5 : 89

Subject 6 : 89

Q. Create a class Book that contains four members: name, author, price and numPages. Include a constructor to set the values for the members. Include methods that could display the complete details of the book. Develop a Java program to create n book objects

⇒ 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.numPages = numPages;

this.price = price;

}

public String toString()

{

String name, author, price, numPages;

name = "Bookname: " + this.Name + "\n";

author = "Authorname: " + this.Author + "\n";

numPages = "Number of pages: " + this.numPages + "\n"

price = "Price: " + this.price + "\n";

return name + author + numPages + price;

}

}

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.print ("Enter the no. of books"),
    n = s.nextInt ();
    Books b[] = new Book[n];
    for (int i = 0; i < n; i++)
    {
        System.out.println ("book");
        System.out.println ("Enter name of book");
        Name = s.nextLine();
        System.out.println ("Enter name of Author");
        Author = s.nextLine();
        System.out.println ("Enter price");
        price = s.nextInt ();
        System.out.println ("Enter num pages");
        numPages = s.nextInt ();
        b[i] = new Books (Name, Author, price, numPages);
    }
    for (int i = 0; i < n; i++)
    {
        System.out.println ("Book" + (i + 1) + "In");
        System.out.println (b[i].toString ());
    }
}
```

O/p

Enter the number of books:

Book 1:

Enter the name of the book : Jungle-Book

Enter the author of the book : Rudyard-Kipling

Enter the price of the book : 1000

Enter the number of pages of the book : 500

19-01-2024

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 class shape. Each one of the classes contain the method printArea() that prints the area of the given shape.

→ import java.util.Scanner;

abstract class shape

{

 double a, b;

 abstract void printArea();

}

class Rectangle extends shape

{

 void PrintArea(double len, double breadth)

 {
 a = len;

 b = breadth;

 double area; // int // typed base and height
 area = a * b;

 System.out.println("Area of rect. is :" + area);

}

}

class Triangle extends shape

{

 void PrintArea(double base, double height)

 {
 a = base; b = height;

 double area = 0.5 * a * b;

 System.out.println("Area of triangle is :" + area);

class circle extends shape

void PrintArea(double radius)

double area;

area = 3.14 * a * a;

y

3

class main

{

public static void main (String [] args)

{

Scanner s = new Scanner (System. in)

- System.out.println ("Enter the length and breadth of rectangle");

double len = s.nextDouble();

double breadth = s.nextDouble();

Rectangle r = new Rectangle ();

- System.out.println ("Enter base and height of triangle");

double base = s.nextDouble();

double height = s.nextDouble();

Triangle t = new triangle ();

- System.out.println ("Enter radius of circle");

double radius = s.nextDouble();

circle c = new circle();

Shape sh; (Object student and student) printing biev

sh = r;

sh.printArea (len, breadth);

sh = t;

sh.printArea (base, height);

sh = c;

Sh. printArea (radius);

O/p

Enter length and breadth of rectangle

5 2

Enter length base and height of triangle

5 10

Enter radius of circle

2.4 5

area of rectangle is : 10

area of triangle is : 25

area of circle is : 78.5

8191124

16/02/2024

Bank Account

⇒ import java.util.Scanner;

class Account {

{

String customerName;

int accNo;

String accType;

double balance;

Account(String customerName, int accNo, String accType,
double balance);

{

this.customerName = customerName;

this.accNo = accNo;

this.accType = accType;

this.balance = balance;

}

void deposit(double amount)

{

balance += amount;

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

}

void displayBalance()

{

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

}

}

class cur-Acc extends Account

{

double min-balance;

double Service-charge;

cur-acct (String customername, int acc-no, double balance, double
min-balance, double service-charge);

{

super (customername, acc-no, "current", balance);

this.min-balance = min-balance;

this.service-charge = Service-charge;

}

void checkMinBalance ()

{

if (balance < min-balance)

{

balance -= Service-charge;

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

}

else

{

System.out.println ("Min. Balance maintained");

}

}

class sav-Acc extends Account

{

double interest-rate;

sav-acct (String customername, int acc-no, double balance, double
interest-rate);

{

System.out.println

Supers (customer name, acc-no, "Savings", balance);

this. interest_rate = interest_rate;

}

Void Deposit Interest()

{

double interest = balance * interest_rate / 100;

balance += interest

System.out.println("interest deposited" + " updated balance" +
balance);

}

void withdraw (double amount)

{

If (amount <= balance)

{

balance -= amount;

System.out.println("remaining balance" + balance);

}

else {

System.out.println("Insufficient balance");

}

}

}

16-02-2024

Public & class Bank

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

```
{  
    Current ca = new Current ("abc", 12345, 1000, 500, 0);
```

```
    ca.deposit(500.0);
```

```
    ca.checkMinBalance();
```

```
    ca.displayBalance();
```

```
Sav acct sa = new Savacct ("xyz", 12367, 2000, 5);
```

```
sa.depositInterest();
```

```
sa.withdraw(300.0);
```

```
sa.displayBalance();
```

```
}
```

```
y
```

Output

Updated Balance : 1500

Min Balance maintained

Balance : 1500

Interest deposited

Updated Balance : 210.0 .

Remaining balance : 1800.0

Balance : 1800.0.

LAB - 5

16-02-2024

a) Packages:

⇒ Package CIE

public class Student

{ public string USN;

public String name;

public int Sem;

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

{

this.USN = USN;

this.name = name;

this.Sem = Sem;

}

public class Internals

{

public int[] marks;

public Internals (int[] marks)

{

this.marks = marks;

}

}

Package SEE

import CIE.Student;

public class External extends Student

{

public int[] marks;

public External (string USN, String name, int Sem, int[] marks)

6-02-2024

```
<
Super (USN, name, Sem);
this. marks = marks;
```

}

}

```
import IIE. Student;
```

```
import SEE. External;
```

```
public class Main
```

<

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

<

```
Student S1 = new Student ("1BN22CS316" "abc", 3);
```

```
int [] SEE. marks = { 86, 80, 92, 94, 89};
```

```
External S2 = new External ("1BN22CS299" "xyz" 3, SEE. marks);
```

```
System.out.println ("Student 1 :");
```

```
System.out.println ("USN " + S1.USN);
```

~~System.out.println ("Name " + S1.name);~~~~System.out.println ("Sem " + S1.Sem);~~~~System.out.println ("Student 2 :");~~~~System.out.println ("USN " + S2.USN);~~~~System.out.println ("Name : " + S2.name);~~~~System.out.println ("Sem : " S2.Sem);~~

}

Output:

Student 1: 1BN22CS316

abc

3

Student 2: 1BN22CS299

xyz

3

16-02-2024

Q2 class wrongAge extends Exception

{

public wrongAge (String str){

Super (str);

}

}

class Father

{

int fAge;

public father (int Age) throws wrongAge

{

if (fAge < 0)

sop ("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;

}

}

16-02-2024.

Public class Main {

 public static void main (String args[]) {

 try {

 Father f = new Father (50);

 Son s = new Son (70, 50);

 }

 catch (WrongAge e) {

 System.out.println (e);

 }

 }

}

Output

wrong age Son's age can't be greater than father's age

16-02-2024

Q3 write a program which creates two threads One thread displaying "BMS college of engineering" once every ten seconds & another displaying "CSE" once every 2 seconds.

=>

```
class BMS implements Runnable {  
    public void run() {  
        while (true) {  
            try {  
                System.out.println("BMS college of engineering");  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
class CSE implements Runnable {
```

```
    public void run() {  
        while (true) {  
            try {  
                System.out.print(" CSE ");  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

16-02-2024

public class Main {

 public static void main (String [] args) {

 Thread t1 = new Thread (new BMS college of Engineering ());

 Thread t2 = new Thread (new CSE ());

 t1.start ();

 t2.start ();

}

}

Output:

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

of creating label, button & 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.add windowListener (this);

Label l = new Label ("Employee Id :");

Button b = new Button ("Submit");

Textfield t = new TextField ();

t.set Bounds (20, 80, 80, 30)

t.set Bounds (20, 100, 80, 30);

b.set Bounds (100, 100, 80, 30);

f.add (b);

f.add (l);

f.add (t);

f.set Size (400, 300);

f.set Title ("Employee info");

f.setLayout (null);

f.setVisible (true);

}

public void windowClosing (WindowEvent e) {

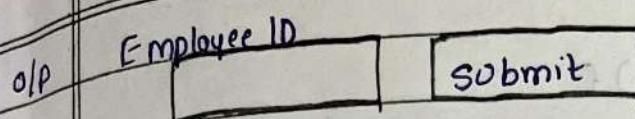
System.exit (0);

public void main (String [] args) {

AWTExample awt_obj = new AWTExample ();

}

y



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 Action Listener {

Frame f;

TextField tf;

Event Handling () {

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) {

if . setText ("welcome");

3

public void windowClosing (WindowEvent e) {

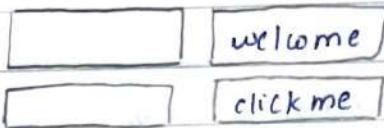
System.exit (0);

3

2 Creating

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

O/P



3 Programs on I/O

1. Example 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 'x' "+ k + " ; special character is: " + ch);  
        }  
    }  
}
```

y

y

y

O/P ASCII Value 35
 ASCII Value 36
 ASCII Value 37
 ASCII Value 38

Example : 2

```
public class FileEx {
    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.print ("Remaining bytes that can be read :" + fin.available ());
        System.out.println ("Remaining bytes that can be read :" + fin.available ());
    }
}
```

O/p Remaining bytes that can be read : 1 A
Remaining bytes that can be read : 0 65

Example : 3

~~```
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 bytes [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)
```

```
}
```

```
}
```

Op no of bytes read : 20

Byte read : Hello it is good.

8 29/2/24