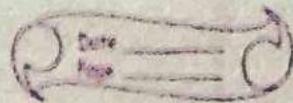


2 January 2024 LAB 2



Book Details

```
import java.util.Scanner;  
class Book {  
    String name;  
    String author;  
    double price;  
    int numPages;  
    Book(String name, String author, double  
        price, int numPages)
```

```
{  
    this.name = name;  
    this.author = author;  
    this.price = price;  
    this.numPages = numPages;  
}
```

void setDetails()

```
{  
    name = name;  
    author = author;  
    price = price;  
    numPages = numPages;  
}
```

void getDetails()

```
{  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter Book Name:  
    name = sc.nextLine();  
    System.out.print("Enter Author name:  
    author = sc.nextLine();  
    System.out.print("Enter Price:  
    price = sc.nextDouble();  
    System.out.print("Enter Number of Pages:  
    numPages = sc.nextInt();  
    Book book = new Book(name, author, price, numPages);  
    book.setDetails();  
    System.out.println("Book Details:  
    System.out.println("Name: " + book.name);  
    System.out.println("Author: " + book.author);  
    System.out.println("Price: " + book.price);  
    System.out.println("Number of Pages: " + book.numPages);  
}
```

author = s. nextline();

System.out.print("Enter Price : ");

price = s. nextDouble();

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

num_pages = s. nextInt();

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

}

public String toString()

{
return ("Book Name :" + name +
"\n Author name :" + author +
"\n Price :" + price + "\n Number
of pages :" + num_pages);

class bookMain {

public static void main(String args) {

{ Scanner s = new Scanner (System.in);
int n, i;

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

n = s.nextInt();

book[] books = new book[n];
for (i=0; i < n; i++)

{ System.out.println("Enter details
of book " +(i+1));

books[i] = new book("", "", 0.0, 0);

books[i].getDetails();

```

    {
        for(i=0; i<n; i++)
        {
            system.out.println ("Enter number of
            books");
            n = sc.nextInt();
            books[i] = new books[n];
            for(i=0; i<n; i++)
            {
                system.out.println ("Enter detail
                of the book " + (i+1));
                system.out.println (books[i]);
            }
        }
    }
}

```

Output

Enter no. of books:

Details of book-1;

Enter the name of book;

Jungle book

Enter name of author:

Rudyard Kipling

Enter the year

200

Enter no. of pages

Details of all books

Name: Jungle book

Author: Rudyard Kipling

Student details

```
import java.util.Scanner;  
class Student {  
    String USN;  
    String name;  
    int marks = 0; int [6];
```

```
void inputDetails()
```

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

```
System.out.print("Enter name");  
name = scanner.nextLine();
```

```
System.out.print("Enter marks of  
6 subjects : ");
```

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

```
{  
    System.out.print("Subject " + (i+1)  
    ": ");  
    marks[i] = scanner.nextInt();
```

```
double calculatePercentage()
```

```
{  
    int totalMarks = 0;
```

```
for (int mark : marks)
```

`totalMarks += mark;`

`} return (double) totalMarks / 6;`

`} void displayDetails()`

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

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

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

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

`public class Student - Det {`

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

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

`System.out.println (" Enter the no. of students: ");`

`int numStudents = scanner.nextInt();`

`Student student[] = new Student [numStudents];`

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

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

`System.out.println (" 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 no. of students : 1

Enter details for student

Enter USN : 123

Enter name : abc

Enter marks for 6 subjects

Subject 1 : 80

Subject 2 : 85

Subject 3 : 90

Subject 4 : 95

Subject 5 : 100

Subject 6 : 90

Details of all subjects

Student Details :

USN : 123

Name : abc

Percentage : 90%

Q1. Write a program to overload the member print that prints sum of 'n' natural numbers when one variable is passed, and prints all prime numbers in a given range when 2 parameters are passed.

any 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;

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

 }

```
class OverloadDemo {  
    public static void main (String args []) {  
        overload o = new Overload (10);  
        o.print (5);  
        o.print (7, 15);  
    }  
}
```

Output Sum of 5 natural numbers is 15
Prime numbers in the range are
7
11

8/12/23

Lab 1

Date _____
Page _____

a) Print Hello World

class Main {

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

```
System.out.println ("Hello World");
```

}

Output : Hello World

b) Adding two numbers

class Main {

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

```
{ int a = 5, b = 4;
```

```
System.out.println (a+b);
```

}

Output : 9

c) Subtracting two numbers

class Main {

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

```
{ int a = 5, b = 4;
```

```
System.out.println (a-b);
```

}

Output : 1

d) Multiplication

class main {

public static void main(String args[])

}

int a = 5, b = 4;

System.out.println(a * b);

}

Output = 20

e) Division

public class main {

public static void main(String args[]) {

int a = 15, b = 3;

System.out.println(a / b);

}

Output = 5

f) Fibonacci Series

public class main {

public static void main(String args[])

int x = 0, y = 1

int next = x + y

System.out.println(x + " " + y)

for (int i = 0; i <= 10; i++)

x = y;

y = next;

next = x + y;

System.out.println(next)

page 10

many real] & there is many map editor.
[many editor-s editor for both
maps & terrain & there many -
output: 0 1 2 3 5 8 13 21 34 58 89

many game engine & editor tool &
many editor tool, engine &
editor with engine very much with
entangled & with editor tool as well

many tools &

((I am part) viewer tools with editor

((I am part) viewer tools with editor)

many tools

viewer-s (part)

(part) viewer

(part) viewer

((I am part) viewer-s (part), viewer)

viewer-s viewer-s with

((I am part) viewer-s with)

LAB 22/01/2024

1. 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 oil, quantity of pulses, and quantity of sugar. You need to return total price. Display the name, phone and total bill of 3 customers

A class Main {

```
public static void main (String args[])
    Grocery g1 = new Grocery ("Shubham", "12345");
    Grocery g2 = new Grocery ("Penny", "32426");
    Grocery g3 = new Grocery ("Howard", "38263");
    g1. totalPrice(20, 33, 56);
    g2. display ();
    g2. totalPrice (23, 24, 144);
    g2. display ();
    g3. totalPrice (1363, 467, 734);
    g3. display ();
}
```

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 totalPhis (double qty_dal, double
 quantity_qty - pulses, double qty_sugar) {
 total = 80 * qty_dal + 50 * qty_pulses
 + 10 * qty_sugar;

}

void display() {
 cout << "Name: ent. printer (" "Name" + " ",
 "Phone Number " + " " + "Jafal");

(CustomerSystem::ent. printer ("C-name" + " " + c-phr
 + " " + total));

}

output	Name	Phone No.	Jafal
Gulberg	12345	3570.0	
Penny	38263	4480.0	
Hawza	34785	59730.0	

:(Jafal) : vehicle

:(Jafal) : vehicle

:(Jafal) : vehicle

:(Jafal) :

:(Jafal) : vehicle

:(Jafal)

2. Write a java program to calculate roots of a quadratic equations. Use appropriate methods to take the input and calculate roots.

A

```
import java.util.Scanner;
```

```
public class QuadraticEquationSolver
```

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

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

```
    double a = s.nextDouble();
```

```
    if (a == 0)
```

```
        System.out.println ("Invalid Input");
```

else

```
{    double b = s.nextDouble();
```

```
    double c = s.nextDouble();
```

```
    double d = b * b - 4 * a * c;
```

```
    if (d > 0)
```

```
        double r1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
        double r2 = (-b - Math.sqrt(d)) / (2 * a);
```

```
        System.out.println ("Two distinct real roots Root 1: " + r1 + ", Root 2: " + r2);
```

```
} else if (d == 0)
```

```
{    double root = -b / (2 * a);
```

```
    System.out.println ("Equal roots, Root" + root);
```

{
dn

{

systems.out.println ("No real roots
exists, complex roots exist.");{ }
}{

{ }

OutputI 2
-5
2

Two distinct real roots : Root 1: 2.0, Root 2: 0.5

II 1
-2
1

Equal roots Root : 1

III 0
3

2

~~invalid input~~

Lab 29.01.2024

B1. Area of all shapes using inheritance

abstract class Shape {

 private int length;

 private int breadth;

Shape (int length, int breadth) {

 this.length = length;

 this.breadth = breadth;

} abstract public void printArea();

} class Rectangle extends Shape {

 Rectangle (int length, int breadth) {

 super (length, breadth);

} public void printArea () {

 System.out.println ("length * breadth")

}

class Triangle extends Shape {

 Triangle (int breadth base, int height) {

 super (base, height);

}

public void printArea () {

 System.out.println ("0.5 * breadth * length");

}

class circle extends Shape
 circle (int radius) {
 super(radius , 0);

}
 public void printArea () {
 System.out.println (3.14 * length
 * length);

public class main {
 public static void main (String args) {

{ rectangle r = new Rectangle (5,5);
 triangle t = new Triangle (3,4);
 circle c = new Circle (5);
 Shape S;
 S = r ;

S.printArea();

S = t ;
 S.printArea();

S = c ;
 S.printArea();

}

20

6.0

50.24

Q2. Develop a java program to create a class bank that maintains two kinds of account for its customers, one called savings account and other 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 and type of account. From this derive the classes cur_acct & sav_acct to make them more specific to this requirement:

- (a) accept deposit from customer & update the balance
- (b) display the balance
- (c) Compute & deposit interest
- (d) Permit withdrawal and update balance
Check for minimum balance, impose penalty if necessary and update the balance

any import java.util.Scanner;

class Account {

protected String customerName;

protected long accountNumber;

protected String accountType;

protected double balance;

public Account (String customerName, long
amountNumber, String amountType,
double balance) {

this. customerName = customerName;
this. amountNumber = amountNumber;
this. amountType = amountType;
this. balance = balance;

} public void displayBalance () {

System.out.print("Account
Number: " + amountNumber);

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

System.out.print("Amount Type: " +
amountType);

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

} public void deposit (double amount) {

balance += amount;

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

} public void withdraw (double amount) {

} if (amount <= balance) {

balance -= amount;

System.out.print("Withdraw of \$" +
amount + " successful"); } }

else {

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

} displayBalance();

}

class Current extends Account {

private double minimumBalance;
private double serviceCharge = 5.0;
public Current (String customerName,
long accountNumber, double balance)
System.out.println (customerName, accountNumber,
"Current", balance);

} public void withdraw (double amount) {

if {

Lab 16.02.2024

Date _____

Page _____

Q Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" & a derived class called "Son" which extends the base class. In Father class, implement a constructor which takes age and throws exception WrongAge() and then the input age ≤ 0 . In Son class, implement a constructor that both Father & Son's age is ≥ 0 and throws exception if Son's age $>=$ Father's age

A // Exception class for wrong age
class WrongAge extends Exception
{ WrongAge() {
super("Invalid age provided");
}}

// Base class Father
class Father {
private int age;
// constructor with exception handling for age
Father(int age) throws WrongAge {
if (age ≤ 0) {
throws new WrongAge();
}
this.age = age;
}}

int getAge() {
return age;

public class main {

 public static void main (String args [])

} try {

 Creating a father instance with
 age 50

 Father father = new Father (50);

 System.out.println ("Father's

 age : " + father.getAge ())

// creating a son instance w/ father's

 age 50 and son's age 25

 Son son = new Son ("Son's age : " + son.

 getAge ());

} catch (WrongAge e) {

 System.out.println (e.getMsg ())

}

 System.out.println ("

 + son.getName ()

 + son.getAge ())

Output

 1. public class main {

 public static void main (String args [])

 2. Father father = new Father (50);

 3. System.out.println ("Father's

 age : " + father.getAge ())

 4. Son son = new Son ("Son's age : " + son.

 getAge ());

 5. } catch (WrongAge e) {

 System.out.println (e.getMsg ())

 6. }

 7. }

 8. }

 9. }

 10. }

 11. }

 12. }

Q Write a program that creates two threads, one thread displaying "BMS College of Engineering" every 10 seconds and another displaying "CSE" once every 2 seconds.

```

class DisplayThread extends Thread {
    private String message;
    private int interval; // interval
    // in millisecond

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

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

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

```

Output

ESE

CSE

CSE

CSE

CSE

BMS : College of Engineering

CSE

CSE

CSE

CSE

BS BMS College of Engineering

✓
X X 3
14.02.21

{ max min max min}

{ max min max min }

max : Don't care (parallel)

min : parallel & equal

max : Don't care (parallel)

min : Don't care (parallel)

Create a package CIE which has 2 classes - Student and Internals. The class Student has attributes such as id, name, gpa. The class Internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import 2 packages in a file that displays the final marks of all students in all five courses.

1. Create a folder cie & save the programs Student.java & Internals.java
2. Create a folder SEE & save the program Internals.java within it
3. Save the main program & outside these 2 folders

25.03.14 (4) Compile Main.java & run the Main. class

~~package CIE;~~ ~~class Main { }~~

~~public class Student {~~
~~String id;~~
~~String name;~~
~~int[] internalMarks = { 12, 15, 18, 19, 20 };~~
~~int[] seeMarks = { 10, 12, 14, 16, 18 };~~

~~package CIE;~~

~~+ " " + id + " " + name + " " + internalMarks[0] + " " + internalMarks[1] + " " + internalMarks[2] + " " + internalMarks[3] + " " + internalMarks[4] + " " + seeMarks[0] + " " + seeMarks[1] + " " + seeMarks[2] + " " + seeMarks[3] + " " + seeMarks[4] + " " + "~~

Score package SSE;
import civ. Student;
public class Cultural extends Student {
 protected int[] SumMarks = new int[5];
}

import CIE. intervals;
import SSE. Cultural;
public class Mani/
 extends Student with marks (String args[])
 Cultural s1 = new intervals();
 s1. age = "123";
 s1. name = "John";
 s1. gpa = 3;
 s1. individualMarks = new int[] { 80, 75,
 90, 85, 88 };

Cultural s2 = new Cultural();

s2. age = "452";

s2. name = "Jane";

s2. gpa = 3;

s2. sumMarks = new int[] {
 85, 78, 92, 88, 90 };

System.out.println "Final marks for "+

s1. name + ":" + calculate

FinalMarks(s1));

Output ("Final marks for " + s2. name +
 ":" + calculateFinalMarks(s2));

{ private static int calculateSumMarks (Student student)

int total = 0;

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

total += (student.internalsMarks[i] +
student.sumMarks[i]) / 2;

}

return total;

} // student sum = ?

(int) sum of internmarks +

(int) average of marks = 3 marks

("marking") writing out = 3 writing

(3 writing, one of 3rd and

(12, 08, 03, 02) sum of 3rd . 3

(2, 08, 001, 02) sum of 3rd . 3

(0, 08, 001, 001) sum of 3rd . 3

(+) 000 . 3

(+) 000 . 3

(+) 000 . 3

(001, 001) writing . 3

"open typeout") all files . 3

(mark) sum of marks . 3

(mark) writing . 3

((sumMarks) private variable then we use

((0) sum value)

((spare part)) we can do it with int sum

Lab 23. 02. 2024

Q1

Create a label, button and JText 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");  
        JTextField t = new JTextField();  
        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[])  
    {  
    }  
}
```

ANTCamponotus ant-obj: no *ANTCamponotus*

* *Drosophila melanogaster*

(2)

output

Employee id - - -

Grobmühle

(Ex. 58, est. 800) show fig. 2.

(cont.) *minigardens* Chap. 3

(+)Jde. + (-)Jde.

(not called up to 2.)

(Shaw) *transact. Soc. Z.*

(cont'd) *Continued.*

10. The following table gives the number of hours per week spent by students in various activities.

Transferred from

(“annuals”) such as

Digitized by srujanika@gmail.com

poorly defined boundaries. It is

10. *Urtica dioica* L. (Urticaceae) - Common Nettle

16.95 ~~16.95~~

—
—
—

(1) 18. 11. 23. 2

1. *gibbons*! *now* *can*

Q2

Create a button and add a action listener for mouse click.

```
import java.awt.*;
import java.awt.event.*;
public class EventHandling extends WindowFrame
    implements ActionListener
{
    Frame f;
    JTextField tf;
    EventHandling()
    {
        f = new Frame();
        f.addWindowListener(this);
        tf = new JTextField();
        tf.setBounds(60, 50, 170, 20);
        JButton b = new JButton("click me");
        b.setBounds(100, 120, 80, 30);
        b.addActionListener(this);
        f.add(b); f.add(tf);
        f.setSize(300, 200);
        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

welcome

click me

2023-07-17