

1. 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 to set and get the details of the objects. Include a toString() method that could display complete details of book (n book objects).

```
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;  
    }  
}
```

```

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.println ("Enter the no. of books : ");
        n = s.nextInt();
        Books b [];
        b = new Books [n];
        b = new
        (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 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) + "\n" + b[i]);
        }
    }
}

```

Output:

Enter the number

Book 1 :

Enter the name

Enter the author

Enter the price

Enter the number

Book 2 :

Enter the name

Enter the author

Enter the price

Enter the number

Book 1 :

Book name

Author :

Price :

Number :

Book 2 :

Book name

Author :

Price :

Number :



Output:

Enter the number of books : 2

Book 1 :

Enter the name of the book : Harry Potter

Enter the author of the book : JK Rowling

Enter the price of the book : 700

Enter the number of pages of the book : 1000

Book 2 :

Enter the name of the book : Percy Jackson

Enter the author of the book : Rick Riordan

Enter the price of the book : 1000

Enter the number of pages of the book : 2000

Book 1 :

Book name : Harry Potter

Author : JK Rowling

Price : 700

Number of pages : 1000

Book 2 :

Book name : Percy Jackson

Author : Rick Riordan

Price : 1000

Number of pages : 2000

2. 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 appr. 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.next();  
        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];  
        }  
        return (double)total / 6;  
    }  
}
```

```
double p = total / 6 ;
```

```
return p ;  
}
```

```
void display()
```

```
{
```

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

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

```
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 Lab1student
```

```
{
```

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

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

```
System.out.println("Enter the number of students:");  
int n = s.nextInt();
```

```
student [ ] students = new student [n];
```

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

```
{
```

~~```
Students [i] = new student ();
```~~~~```
System.out.println ("\nEnter details of student " + (i+1) + ":")
```~~~~```
Students [i].Details ();
```~~~~```
{
```~~

```
for (student student : students)
```

```
{
```

```
student.display ();
```

```
{
```

```
{
```

Output:

Enter the number of Students : 1

Enter details of student 1 :

Enter USN : 123

Enter name : Hadiya F

Enter marks for 6 subjects

Subject 1 : 23

Subject 2 : 45

Subject 3 : 90

Subject 4 : 80

Subject 5 : 70

Subject 6 : 75

Student details

USN : 123

Marks :

3. Quadratic Equations

```
import java.util.Scanner;  
import java.lang.Math;  
class quadratic  
{ public static void main (String [] args)  
{  
    int a,b,c ;  
    System.out.println ("Enter the values of a, b, c resp");  
    Scanner sc = new Scanner (System.in);  
    a = sc.nextInt();  
    b = sc.nextInt();  
    c = sc.nextInt();  
    Equation eq = new Equation (a,b,c);  
    eq.quad ();  
}}
```

Class Equation

```
int a, b, c ;  
Equation (int x, int y, int z);  
{  
    a = x ;  
    b = y ;  
    c = z ;  
}  
quad ()  
{
```

~~if ($a=0$)
 System.out.println ("Not quadratic eq");
else if ($a>0$)~~

{
Sopln ("Two real and different solution");

double $x_1 = -b + \text{Math.sqrt}((d) / (2 * a))$;

$\text{sqrt}((d) / (2 * a))$;

double $x_2 = -b - \text{Math.sqrt}((d) / (2 * a))$;

Sopln (x_1);

Sopln (x_2);

}

else if ($d == 0$)

{

Sopln ("Real & same roots");

$x_1 = -b + \text{Math.sqrt}(d / (2 * a))$;

$x_2 = -b - \text{Math.sqrt}(d / (2 * a))$;

Sopln (x_1);

Sopln (x_2);

}

else if ($d < 0$)

{

Sopln ("Imaginary Solutions");

}

}

Output:

Enter the values of a, b, c

2

3

2

Imaginary solutions

for
8 // 2j

CAB - 2

1. Develop a java program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide 3 classes , rectangle, triangle, circle extending shape . & print area of the shape.
- abstract class Shape

{

```
protected int value1;  
protected int value2;  
public Shape (int value1, int value2)  
{ this.value1 = value1;  
this.value2 = value2; }
```

```
public abstract void printArea();  
}
```

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

```
public void printArea () {  
int area = value1 * value2  
System.out.println (" Area of rectangle is : " + area);  
}  
}
```

```
class triangle extends shape {
```

```
public triangle (int base, int height)  
{ super (base, height); }
```

~~```
public void printArea () {
double area = 0.5 * value1 * value2;
System.out.println (" Area of triangle is : " + area);
}
```~~

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

```
public void printArea ()
{ double area = Math.PI * value1 * value1;
 System.out ("Area of circle is : " + area);
}
```

```
public class Hadia {
 public static void main (String [] args)
 { Rectangle rectangle = new Rectangle (5, 7);
 rectangle.printArea ();
 Triangle triangle = new Triangle (5, 8);
 triangle.printArea ();
 Circle circle = new Circle (8);
 circle.printArea ();
 }
}
```

Output.

Area of rectangle is : 35

Area of triangle is : 20.0

Area of circle is : 201.061929829574676

2.

Class Bank {

psvm (String [] args)

{ SavingsAccount savingsAccount = new

SavingsAccount ("Diksha N", "SA1001");

CurrentAccount currentAccount = new

CurrentAccount ("Hanuitha B", "CA2002");

savingsAccount . deposit (5000);

savingsAccount . displayBalance ();

savingsAccount . computeInterest ();

SavingsAccount . displayBalance ();

savingsAccount . withdraw (2000);

savingsAccount . displayBalance ();

currentAccount . deposit (8000);

currentAccount . displayBalance ();

currentAccount . withdraw (5000);

currentAccount . displayBalance ();

3

3

Class Account {

protected String customerName;

protected String accountNumber;

protected double balance;

public Account (String customerName, String accountNumber)

{

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = 0; }

public void deposit (double amount)

{ balance += amount;

SOP ("Deposit of \$" + amount + "successful");

}

public void displayBalance ()

{ SOP ("Account number:" + accountNumber  
+ "\nBalance : \$" + balance);

}

}

class SavingsAccount extends Account {

public SavingsAccount (String customerName,  
String accountNumber)

{ super (customerName, accountNumber); }

public void computeInterest ()

{ double interestRate = 0.05;

double interest = balance \* interestRate;

balance += interest;

SOP ("Interest of \$" + interest + " computed

and added to the balance."); }

```
public void withdraw (double amount)
{ if (balance >= amount)
 { balance -= amount;
 SOP ("Withdrawal of $" + amount + " successful.");
 }
```

else

```
{ SOP ("Insufficient funds");
}
```

}

}

```
class CurrentAccounts extends Account
```

```
{ private double minimumBalance = 1000;
```

```
public CurrentAccount (String customerName,
String accountNumber)
```

```
{ super (customerName, accountNumber); }
```

```
public void withdraw (double amount)
```

```
{ if (balance - amount >= minimumBalance)
```

```
{ balance -= amount;
```

```
SOP ("Withdrawal of $" + amount + " successful.");
}
```

else

```
{ SOP ("Insufficient funds");
}
```

```
imposeServiceCharge ();
}
```

3

private void imposeServiceCharge ()

```
{ double serviceCharge = 20;
```

```
balance -= serviceCharge;
```

```
SOP ("Service of $" + serviceCharge + " imposed.");
}
```

3

Output:

Deposit of 5000 successful

Account number : SA1001

Balance : 5000

Interest of 250 computed & added to balance

Balance : 5250

Withdrawal of \$ 2000 successful

Acc. no : SA1001

Balance = 3250

Deposit of 8000 successful

Acc no : CA2002

Sal : 8000

Withdrawal of 8000 successful

Acc no : CA2002

Balance : 3000.0

Done  
29/1/24

LAB - 4

Package Program :

CIE → Student.java &amp; Internals.java

SEE → External.java

outside the folders main.java.

Student.java →

```
package cie;
public class Student {
 public String name;
 public String usn;
 public int Sem;
}
```

Internals.java →

```
package cie;
public class Internals extends Student {
 public int [] marks = new int [5];
}
```

External.java →

```
package see;
import cie.Student;
public class External extends Student {
 public int [] seemarks = new int [5];
}
```

Main.java →

```
import cie.internals;
import see.external;
import java.util.Scanner;

public class Main {
 psvm (String [] args) {
 sopln ("Enter the no. of students");
 Scanner input = new Scanner (System.in);
 int n = input.nextInt();
```

```
 Internals [] S1 = new Internals [n];
 External [] S2 = new External [n];
 int [] finalcie = new int [n];
 int [] finalsee = new int [n];
```

```
 for (int i = 0; i < n; i++)
 {
 S1 [i] = new Internals ();
 sopln ("Enter the name");
 S1 [i].name = input.next();
 sopln ("Enter the usn");
 S1 [i].usn = input.next();
 sopln ("Enter the sem");
 S1 [i].sem = input.nextInt();
 sopln ("Enter the marks of 5 subjects");
 for (int j = 0; j < 5; j++)
 {
 S1 [i].marks [j] = input.nextInt();
 finalcie [i] += S1 [i].marks [j];
 }
 }
```

```
for (int i=0 ; i<n ; i++)
{
 S2[i] = new External();
 SOPIn ("Enter the name");
 S2[i].name = input.next();
 SOPIn ("Enter the usn");
 S2[i].usn = input.next();
 SOPIn ("Enter the sem");
 S2[i].sem = input.nextInt();
}

SOPIn ("Enter the marks of 5 subjects");
for (int j=0 ; j<5 ; j++)
{
 S2[i].seemarks[j] = input.nextInt();
 finalsee[i] += S2[i].seemarks[j];
}
}

SOPIn ("Final Marks :");
for (int i=0 ; i<5 ; i++)
{
 SOPIn ("Name : " + S1[i].name +
 " USN : " + S1[i].usn +
 " Sem : " + S1[i].sem);
}

SOPIn ("Internal marks : " + finalcie[i]);
SOPIn ("External marks : " + finalsee[i]);
SOPIn ("Total marks : " + (finalcie[i] + finalsee[i]));
}
```

pg

Date

Output

Enter the no. of students

1

Enter the name

Jack

Enter the USN

1BM44

Enter the Sem

3

Enter the marks of 5 subjects

20

20

20

20

20

Enter the name

Jack

Enter the USN

1BM44

Enter the sem

3

Enter the marks of 5 subjects

10

10

10

10

10

Final marks :

Name : Jack USN : 1BM44

Sem : 3

Internal marks : 100

External marks : 50

Total Marks : 150

Jack  
90/100

LAB - 5

1. Write a program to demonstrate exception handling : Create base class "father" and a derived class "Son" which extends base class. In father class, implement a constructor which takes the age and throws the exception WrongAge() when input age < 0. In Son class, implement a constructor that takes both father & sons age and throws an exception if sons age  $\geq$  fathers age.

```
class WrongAge extends Exception
{ public WrongAge (String message)
 { super (message); }
```

```
class Father {
 int age;
 public Father (int age) throws WrongAge {
 if (age < 0)
 { throw new WrongAge ("Age cannot be negative"); }
 this.age = age; }

 public int getAge()
 { return age; }}
```

```
class Son extends Father
{ int sonAge;
 public Son (int fatherAge, int sonAge) throws WrongAge
 { super (fatherAge);
 if (sonAge \geq fatherAge)
 { throw new WrongAge ("Son's age cannot be greater than or
 equal to father's age"); }
 this.sonAge = sonAge; }
```

Date: / /

```
public class CheckAge {
 public void psvm (String [] args) {
 try {
 Father father = new Father (10);
 System.out ("Father's age : " + father.getAge ());
 Son son = new Son (10, 15);
 System.out ("Son's age : " + son.getSonAge ());
 } catch (WrongAge e) {
 System.out ("Exception caught : " + e.getMessage ());
 }
 }
}
```

Output :

Father's age : 10

Exception Caught : Son's age cannot be greater than  
or equal to father's age.

2. WAP which creates two threads , one thread displaying "BMS College of Engineering" once every 10 seconds and another displaying "CSE" once every 2 seconds.

```
class DispMessage extends Thread
{ String message ;
 int interval ;
```

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

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

```
Public class BMSCSE {
 public static void main (String [] args)
{ DispMessage bmsThread = new DispMessage ("BMS college of
Engineering" , 10000) ;
 DispMessage cseThread = new DispMessage ("CSE" , 2000) ;
 bmsThread . start () ;
 cseThread . start () ;
}}
```

Output :

BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE

BMS College of Engineering

Date  
19/2/24

papergrid  
Date

### Lab 6 :

AWT      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();
 f.setBounds(20, 80, 80, 30);
 t.setBounds(20, 100, 80, 30);
 b.setBounds(100, 100, 80, 30);
 f.add(t);
 f.add(l);
 f.add(b);
 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();
 }
}
```

papergrid  
Date 26/2/23

## Programs on IO

```
import java.io.
public class ByteArrayInput

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.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(); }
}
```

papergrid  
Date: / /

## 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 bt = new ByteArrayInputStream(buf);
 int k = 0;
 while ((k = bt.read()) != -1) {
 char ch = (char) k;
 System.out.println("ASCII value of character is: " + k + "
 ; Special character is: " + ch);
 }
 }
}
```

### 2. Example 2

```
import java.io.*;
public class ByteArrayInput {
 public static void main(String[] args) throws IOException
 { byte[] buf = {35, 36, 37, 38};
 ByteArrayInputStream bt = new ByteArrayInputStream(buf);
 int k = 0;
 while ((k = bt.read()) != -1) {
 char ch = (char) k;
 System.out.println("ASCII value of character is: " + k + "
 ; Special character is: " + ch);
 }
 }
}
```

papergrid  
Date: / /

### 3. Example 3

```
public class FileEx {
 psvm (String a[]) throws IOException
 { FileInputStream fin = new FileInputStream ("Example.txt");
 int content;
 sopln ("Remaining bytes that can be read : " +
 fin.available());
 }
}
```

Content = fin.read();

sop ((char) content + " ");

sop (content + " ");

```
sopln ("remaining bytes that can be read : "
 + fin.available());
```

```
sopln ("remaining bytes that can be read : "
 + fin.available());
```

}

}

### 4. Example 4.

```
import java.io.FileInputStream;
import java.io.IOException;
public class FileEx2 {
 psvm (String a[]) throws IOException {
 FileInputStream fin = new FileInputStream ("Example.txt");
 byte [] bytes = new byte [20];
 int i; char c; i = fin.read (bytes);
 }
}
```

sopln ("No. of bytes read : " + i);

sop ("Bytes read : ");

```
for (byte b : bytes) {
 c = (char) b;
```

```
 sop (c); }
```

}

