

```
{
```

```
    System.out.println(n + " is prime");
```

```
}
```

```
}
```

```
}
```

```
}
```

output: 3 is a prime number

11. Groceries using ~~construction~~

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

```
class groceries
```

```
{
```

```
    double dal, sugar, pulses;
```

```
    groceries()
```

```
{
```

```
        dal = 1;
```

```
        pulses = 1;
```

```
        sugar = 0.5;
```

```
}
```

```
    groceries (double a)
```

```
{
```

```
        dal = pulses = sugar = a;
```

```
}
```

```
    groceries (double a, double b, double c)
```

```
{
```

```
        dal = a;
```

```
        pulses = b;
```

```
        sugar = c;
```

```
}
```

```
    groceries (groceries ob)
```

```
{
```

```

class circle extends shape {
    public Circle (int radius) {
        this.width = radius;
        this.height = radius;
    }
}

```

```

    public void printArea()
    {
        System.out.println(" Circle Area: " +
            (Math.PI * width * height));
    }
}

```

```

public class stack {
    public static void main (String[] args)
    {
        shape rectangle = new Rectangle(5,3);
        shape triangle = new Triangle(4,6);
        shape circle = new circle(2);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

output:

Enter length and width of rectangle
2
3

Area of Rectangle: 6.0 square units.

Enter height and base of Triangle
5
3

Area of Triangle 7.5 square units.

Let radius of circle is r
 Area of circle is πr^2 square unit

19-02-24

3) Develop a Java program to create class an abstract class named Shape that contains two integer and an empty method named printArea()

```
abstract class Shape {
    protected int width, height;

    public abstract void printArea();
}
```

```
class Rectangle extends Shape {
    public Rectangle (int width, int height) {
        this.width = width;
        this.height = height;
    }
```

```
    public void printArea() {
        System.out.println("Rectangle Area" +
            (width * height));
    }
}
```

```
class Triangle extends Shape {
    public Triangle (int base, int height) {
        this.width = base;
        this.height = height;
    }
```

```
    public void printArea() {
        System.out.println("Triangle Area" +
            (0.5 * width * height));
    }
}
```

```

Thread.sleep(2000);
}
catch (InterruptedException e)
{
}
}
}
}

```

```

public class Threads
{
    public static void main(String args[])
    {
        Thread t1 = new Thread1();
        Thread t2 = new Thread2();
        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

```

catch (WrongAge e)

1

System.out.println("Exception: " +
e.getMessage());

}

}

}

output:

①

Enter Father's age:

40

Enter son's age:

25

Father's age: 40

Son's age: 25

②

Enter Father's age:

80

Enter son's age:

55

Exception: son's age cannot be greater than or equal to Father's age

Enter Father's age: 40

③

Enter son's age:

25

Father's age: 40

son's age: 25

③

Enter Father's age:

50

Enter son's age:

55

Exception: son's age cannot be greater than or equal to Father's age.

// Threads

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

```
class Thread1 extends Thread
{
    public void run()
    {
        while (true)
        {
            System.out.println("BMS  
College of Engineering");
            try
            {
                Thread.sleep(10000);
            }
            catch (InterruptedException e)
            {
            }
        }
    }
}
```

```
class Thread2 extends Thread
{
    public void run()
    {
        while (true)
        {
            System.out.println("CSE");
            try
            {
                Thread.sleep(2000);
            }
            catch (InterruptedException e)
            {
            }
        }
    }
}
```

// Exception Handling.

- Q) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class call "son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0 . In son class, implement a constructor that takes both father and son's age and throws an exception if son's age is \geq father's age.

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

```
class WrongAge extends Exception {
```

```
    public WrongAge() {
```

```
        super("Age Error");
```

```
    }
```

```
    public WrongAge(String message) {
```

```
        super(message);
```

```
    }
```

```
}
```

```
class InputScanner {
```

```
    protected Scanner scanner;
```

```
    public InputScanner() {
```

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

```
    }
```

```
    public int nextInt() {
```

```
        return scanner.nextInt();
```

```
    }
```

```
}
```

```
}
```


3

}

2

h

try 2

'son son = new soul')

system.out.println("Father's age:
+ son. ~~father~~ Age);

System.out.println ("son's age: " + son.sonAge);

3

catch (WrongAge e)

```
System.out.println("Exception: " +
    e.getMessage());
```

```
}
```

```
}
```

Output:

Enter Father's age:

40

Enter son's age:

25

Father's age: 40

son's age: 25

② Enter Father's age:

50

Enter son's age:

55

Exception: son's age cannot be greater than or equal to father's age

Enter Father's age: 40

② Enter son's age:

25

Father's age: 40

son's age: 25

③ Enter Father's age:

50

Enter son's age:

55

Exception: son's age cannot be greater than or equal to father's age

```
// package SEE;
import CIE. student;

public class External extends CIE. student
{
    public int [] SeeMarks;

    public External (String name, String usn,
        int sem, int [] See Marks)
    {
        super (name, usn, sem);
        this. See Marks = See Marks;
    }
}
```

```
// Public class FinalMarked
public static void main (String [] args) {
    Scanner s1 = new Scanner (System.in);
    System.out.println("Enter the
        number of students");
    int n = s1.nextInt();

    String [] name = new String [n];
    String [] usn = new String [n];
    int [] sem = new int [n];
    int [][] InternalMarks = new int [n][5];
    int [][] See Marks = new int [n][5];

    for (int i = 0; i < n; i++)
    {
        System.out.println("Enter Details of
            Student " + (i+1) + " :");
        System.out.println("In Name:");
        name[i] = s1.next();
    }
}
```



```

System.out.println("\nUSN:");
usn[i] = sl.next();
System.out.println("\nSem:");
sem[i] = sl.nextInt();

System.out.println("Enter Internal
marks for 5 courses:");
for (int j = 0; j < 5; j++)
{
    System.out.println("In Course" + (j+1) + ":");
    InternalMarks[i][j] = sl.nextInt();
}

System.out.println("Enter See marks for
5 courses:");
for (int j = 0; j < 5; j++)
{
    System.out.println("In Courses" + (j+1)
        + ":");
    SeeMarks[i][j] = sl.nextInt();
}

int [][] finalMarks = new int [n][5];
for (int i = 0; i < n; i++)
{
    Internal i1 = new Internals (name[i], usn[i],
        sem[i], InternalMarks[i]);
    External e1 = new Internals (name[i],
        usn[i], sem[i], SeeMarks[i]);
    int sum = 0;
    for (int j = 0; j < 5; j++)
    {
        finalMarks[i][j] = i1.InternalMarks[i]
            + e1.SeeMarks[j];
    }
}

```

```

System.out.println("In\nFinal marks for"
+ n+" student on course");
for (int k=0; k<n; k++)

```

```

    System.out.println(name[k]+":");
    for (int j=0; j<5; j++)

```

```

        sum += final marks [j][j];
    }

```

```

} System.out.println(sum);

```

```

}
} st.close();

```

or

```

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

```

```

    SOP (name[k]+":");

```

```

    SOP ("CIE marks:"+"");

```

```

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

```

```

        S.O.P ("course" + (j+1) + ":" +
        Internal marks [k][j]);
    }

```

```

    SOP ("SEE marks:"+"");

```

```

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

```

```

        SOP ("course" + (j+1) + ":" +
        See Marks [k][j]);
    }

```

```

    SOP ("\n\n");

```

```

}

```

```

} st.close();

```

output:

Enter the number of students
1

Enter details of student:

Name: Neha

USN: 074

Sem: 3

Enter internal marks for 5 courses:

Course 1: 33

Course 2: 36

Course 3: 40

Course 4: 45

Course 5: 70

Enter see marks for 5 courses:

Course 1: 45

Course 2: 32

Course 3: 37

Course 4: 80

Course 5: 50

Final marks for 1 student in 5 courses:

536.

29/1/24


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

```
Account account;
```

```
if (accountType.equals("Savings")) {  
    account = new SavingsAccount(  
        customerName, accountNumber, account  
        Type, balance);  
}
```

```
else
```

```
{
```

```
    account = new CurrentAccount(customer  
        Name, account, accountType, balance);
```

```
while (True) {
```

```
    system.out.println(" 1. Deposit");
```

```
    system.out.println(" 2. Display balance");
```

```
    system.out.println(" 3. Compute and  
        deposit interest");
```

```
    system.out.println(" 4. Withdraw");
```

```
    system.out.println(" 5. Exit");
```

```
    system.out.println(" Enter choice:");
```

```
    int choice = s.nextInt();
```

```
switch (choice)
```

```
{
```

```
    case 1: system.out.print(" Enter amount  
        to deposit: ");
```

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

```
        account.deposit(amount);
```

```
        break;
```

```
    case 2: account.displayBalance();
```

```
        break;
```

case 3:

```

if (account instanceof SavingsAccount)
{
    ((SavingsAccount) account).compoundInterest();
}

```

else

{

```

    System.out.println("Interest not available for current account");
    break;
}

```

case 4:

```

System.out.println("Enter amount to withdraw:");

```

```

Amount = S.nextDouble();

```

```

Amount = withdraw (Amount);

```

```

if (Account instanceof CurrentAccount)
{
    ((CurrentAccount) account).imposeServiceCharge();
}

```

```

break;

```

case 5:

```

System.exit(0);

```

```

}

```

```

}

```

```

}

```

```

}

```

// Packages

classmate

Date _____
Page _____

Folder Structure

Folder : cse → Internal.java, Student.java
 see → external.java

// package cse;

public class Internal extends CIE

{

 public int[] InternalMarks;

 public Internal(String name,
 String usn, int sem, int[] InternalMarks)

{

 super(name, usn, sem);

 this.InternalMarks = InternalMarks;

 }

}

// package cse;
public class Student

{

 String name;

 String usn;

 int sem;

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

{

 this.name = name;

 this.usn = usn;

 this.sem = sem;

 }

}

Book 2:

Enter name of book: Name 2

Enter name of author: Author 2

Enter Price: 200

Enter Pages: 200

Book 1:

Book name: Name 1

Author name: Author 1

Number of Pages: 100

Price: 100

Book 2:

Book name: Name 2

Author name: Author 2

Number of Pages: 200

Price: 200

Bank.

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

```
class Account {
```

```
    String customerName;
```

```
    int accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    Account (String customerName, int  
            accountNumber, String accountType,  
            double balance)
```

```
    {
```

```
        this.customerName = customerName;
```

```
        this.accountNumber = accountNumber;
```

```
        this.accountType = accountType;
```

```
        this.balance = balance;
```

```
    }
```

```
    void deposit (double amount) {
```

```
        balance += amount;
```

```
        System.out.println("Balance=" + balance); } }
```

```
    void displayBalance () {
```

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

```
    }
```

```
    void withdraw (double amount) {
```

```
        if (balance - amount < 0) {
```

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

```
            return;
```

```
        }
```

```
        balance -= amount;
```

```
        System.out.println("Withdrawal of " +
```

```
        amount + " successful")
```

```

class SavingsAccount extends Account {
    SavingsAccount (String customerName,
    int accountNumber, String accountType,
    double balance),

```

```

{

```

```

    super (customerName, accountNumber,
    accountType, balance);
}

```

```

void compoundInterest()
{

```

```

{

```

```

    double rate = 0.05;

```

```

    double time = 1.0;

```

```

    double interest = balance * Math.pow
    (1 + rate, time) - balance;

```

```

    balance = amount;

```

```

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

```

```

}

```

```

class CurrentAccounts extends Account {
    double minimumBalance = 1000;
    double serviceCharge = 50;

```

```

    CurrentAccount (String customerName,
    int accountNumber, String accountType,
    double balance)

```

```

{

```

```

    super (customerName, accountNumber,
    accountType, balance);
}

```



```
void withdraw(double amount) {  
    if (balance - amount < minimumBalance) {  
        System.out.println("Insufficient  
        balance");  
        return; }  
    balance -= amount;  
    System.out.println("Withdrawal of " +  
    amount + " successful");  
}
```

```
void imposeServiceCharge() {  
    if (balance < minimumBalance) {  
        balance += serviceCharge;  
        System.out.println("Service charge of " +  
        service + " imposed");  
    }  
}
```

```
class Bank {  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
        System.out.print("Enter customer  
        name: ");  
        String customerName = s.nextLine();  
        System.out.print("Enter account  
        number:");  
        int accountNumber = s.nextInt();  
        System.out.print("Enter account type  
        (savings / current):");  
        String accountType = s.next();  
        System.out.println("Enter initial  
        balance:");  
    }  
}
```

```
class main
```

```
{
```

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

```
    {
```

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

```
        int n;
```

```
        String name;
```

```
        String author;
```

```
        int price;
```

```
        int numPages;
```

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

```
        n = sc.nextInt();
```

```
        Books b[];
```

```
        b = new Books [n];
```

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

```
        {
```

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

```
            System.out.print("Enter name of books:");
```

```
            name = sc.next();
```

```
            System.out.print("Enter author:");
```

```
            author = sc.next();
```

```
            System.out.print("Enter price:");
```

```
            price = sc.nextInt();
```

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

```
            numPages = sc.nextInt();
```

```
            b[i] = new Books (name, author, price,  
            numPages);
```

```
        }
```

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

```
        {
```

```
            System.out.println("Book" + (i+1) + " : (" + b[i].name + ", " + b[i].author + ", " + b[i].price + ", " + b[i].numPages + ")");
```

```
        }
```

```
}
```


Output:

Enter the number of books : 2

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

Book 2:

Enter the name of the book :

Tales - of - Akbar - And - Birbal

Enter the author of the book : Birbal

Enter the price of the book : 900

Enter the number of pages of the book : 400

Book 1:

Book name : Jungle - Book

Author : Rudyard-Kipling

Price : 1000

Number of pages : 500

Book 2:

Book name : Tales - of - Akbar - And - Birbal

Author : Birbal

Price : 900

Number of pages : 400.

Create a class Book that contains four members: name, author, price and ~~name~~ - pages. Include a constructor to set the values for the members.

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, author;  
    int price, num-pages;  
    Book (String name, String author, int  
    price, int num-pages)  
    {
```

```
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num-pages = num-pages;  
    }
```

```
    public String toString()  
    {
```

```
        String name, author, price, num-pages;  
        name = "Book name: " + this.name + "\n";  
        author = "Author name: " + this.author + "\n";  
        num-pages = "Number of pages: " + this.num-  
            pages + "\n";  
        price = "Price: " + this.price + "\n";  
        return name + author + num-pages +  
            price;  
    }
```

```
}
```

```
class Main {
```

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

```
        Scanner s = new Scanner (System.in);  
        int n;  
        String Name;  
        String Author;  
        int price;
```

```

int num-pages;
System.out.println("Enter the number of
books;");
n = s.nextInt();
Books b[] = new Books[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 name of author");
    author = s.next();
    System.out.println("Enter Price:");
    price = s.nextInt();
    System.out.println("Enter Page:");
    numPages = s.nextInt();

```

```

    b[i] = new Books (name, author,
        price, num-pages);
}

```

```

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

```

```

{
    System.out.println("Books" + (i+1) + "\n");
    System.out.println(b[i].toString());
}

```

output:

Enter the number of books : 2

Books :

Enter name of book : Name 1

Enter name of author : Author 1

Enter Price : 100

Enter Pages : 100

```
dal = ob.dal;
pulses = ob.pulses;
sugarr = ob.sugarr;
}

void cal_amt()
{
    System.out.println(dal * 50 +
        pulses * 80 + sugarr * 100);
}
}

class Run
{
    public static void main(String args[])
    {
        double a, b, c;
        JOptionPane.showMessageDialog(
            "Enter values:");
        groceries g1 = new groceries();
        Scanner sc = new Scanner(System.in);
        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();
        groceries g2 = new groceries(a);
        groceries g3 = new groceries(a, b, c);
        groceries g4 = new groceries(g3);

        g1.cal_amt();
        g2.cal_amt();
        g3.cal_amt();
        g4.cal_amt();
    }
}
```



```

else if (d == 0)

```

```

    x1 = x2 = -b / (2 * a);

```

```

    System.out.println("The roots are  
real and equal:" + x1 + "and" + x2);

```

```

}

```

```

}

```

```

}

```

```

class Quadran

```

```

{

```

```

    public static void main (String args[])
    {

```

```

        Quad q = new Quad();

```

```

        q.input();

```

```

        q.calculate();
    }

```

```

}

```

output: Enter the numbers:

Enter a: 10

Enter b: 1

Enter c: 1

The roots are imaginary & distinct

// Quadratic

import java.util.*;

import java.lang.*;

class Quad

int a;

int b;

int c;

void input()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the numbers");

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

a = sc.nextInt();

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

b = sc.nextInt();

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

c = sc.nextInt();

}

void calc()

{

float d, x1, x2;

d = (b * b - (4 * a * c));

if (d > 0)

{

x1 = (float) (-b - Math.sqrt(b * b - 4 * a * c) / (2 * a));

x2 = (float) (-b + Math.sqrt(b * b - 4 * a * c) / (2 * a));

System.out.println("The roots are imaginary and distinct");

System.out.println("root 1: " + x1 + " + i" + x2);

System.out.println("root 2: " + x1 + " - i" + x2);

}

```
//Book database.
```

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

```
class Books
```

```
{
```

```
    String name;
```

```
    String author;
```

```
    int price;
```

```
    int numPages;
```

```
    Book()
```

```
    {
        String name = "";
```

```
        String author = "";
```

```
        int price = 0;
```

```
        int numPages = 0;
```

```
    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;
```

```
}
```