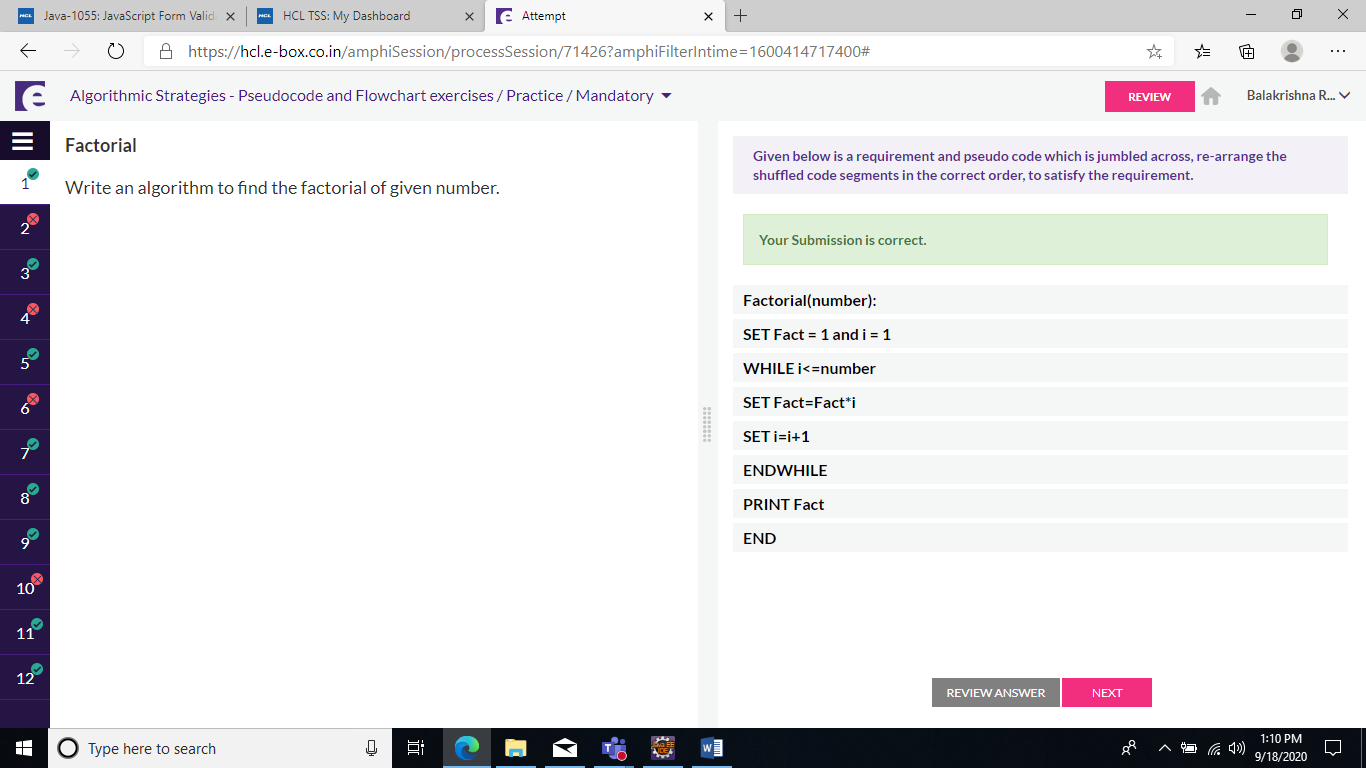
**- Programming Fundamentals**



Write an algorithm to find the factorial of given number.

**Factorial(number):**

**SET Fact = 1 and i = 1**

**WHILE i<=number**

**SET Fact=Fact\*i**

**SET i=i+1**

**ENDWHILE**

**PRINT Fact**

**END**

**Fibonacci Sequence**  
  
Write an algorithm to generate the Fibonacci Sequence upto to the given number.

S3P16-Series1

**Series 1**

The Event Organizing Company "Buzzcraft" focuses event management in a way that creates a win-win situation for all involved stakeholders. Buzzcraft don't look at building one time associations with clients, instead, aim at creating long-lasting collaborations that will span years to come. This goal of the company has helped them to evolve and gain more clients within notable time.  
The number of clients of the company from the start day of their journey till now is recorded sensibly and is seemed to have followed a specific series like: 2,3,5,7,11,13,17,19, 23 ...  
   
Write a program which takes an integer N as the input and will output the series till the Nth term.  
   
**Input Format:**  
First line of the input is an integer N.  
  
**Output Format:**  
Output a single line the series till Nth term, each separated by a comma.  
Refer sample input and output for formatting specifications.  
  
**Sample Input 1:**  
5  
  
**Sample Output 1:**  
2 3 5 7 11  
  
**Sample Input 2:**  
10  
  
**Sample Output 2:**  
2 3 5 7 11 13 17 19 23 29

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int a = Integer.parseInt(s.nextLine());

int ct=0,n=0,i=1,j=1;

while(n<a) {

j=1;

ct=0;

while(j<=i) {

if(i%j==0){

ct++;

}

j++;

}

if(ct==2) {

System.out.printf("%d ",i);

n++;

}

i++;

}

}

}

**Total Expenses for the Event**

The prime functionality of an Event Management System is budgeting. An Event Management System should estimate the total expenses incurred by an event and the percentage rate of each of the expenses involved in planning and executing an event. Nikhil, the founder of "Pine Tree" wanted to include this functionality in his company’s Amphi Event Management System and requested your help in writing a program for the same.  
   
The program should get the branding expenses, travel expenses, food expenses and logistics expenses as input from the user and calculate the total expenses for an event and the percentage rate of each of these expenses.  
   
**Input Format:**  
First input is a int value that corresponds to the branding expenses.  
Second input is a int value that corresponds to the travel expenses.  
Third input is a int value that corresponds to the food expenses.  
Fourth input is a int value that corresponds to the logistics expenses.  
   
**Output Format:**  
First line of the output should display the int value that corresponds to the total expenses for the Event.  
Next four lines should display the percentage rate of each of the expenses.  
Refer sample input and output for formatting specifications.  
**[All text in bold corresponds to input and rest corresponds to output.]**  
  
**Sample Input and Output:**  
Enter branding expenses  
**20000**  
Enter travel expenses  
**40000**  
Enter food expenses  
**15000**  
Enter logistics expenses  
**25000**  
Total expenses : Rs.100000.00  
Branding expenses percentage : 20.00%  
Travel expenses percentage : 40.00%  
Food expenses percentage : 15.00%  
Logistics expenses percentage : 25.00%

Top of Form

Bottom of Form

**Additional Sample TestCases**

**Sample Input and Output 1 :**

Enter branding expenses

855

Enter travel expenses

877779

Enter food expenses

5544

Enter logistics expenses

2256

Total expenses : Rs.886434.00

Branding expenses percentage : 0.10%

Travel expenses percentage : 99.02%

Food expenses percentage : 0.63%

Logistics expenses percentage : 0.25%

import java.util.Scanner;

import java.text.DecimalFormat;

class Main {

public static void main(String[] args) {

DecimalFormat df = new DecimalFormat("0.00");

Scanner sc = new Scanner(System.in);

System.out.println("Enter branding expenses");

int branding = sc.nextInt();

System.out.println("Enter travel expenses");

int travel = sc.nextInt();

System.out.println("Enter food expenses");

int food = sc.nextInt();

System.out.println("Enter logistics expenses");

int logistics = sc.nextInt();

double totalexpense = branding+food+travel+logistics;

double brandingper=branding\*100/totalexpense;

double travelingper=travel\*100/totalexpense;

double foodper=food\*100/totalexpense;

double logisticsper=logistics\*100/totalexpense;

System.out.println("Total expenses : Rs."+df.format(totalexpense));

System.out.println("Branding expenses percentage : "+df.format(brandingper)+"%");

//System.out.print("% \n");

System.out.println("Travel expenses percentage : " +df.format(travelingper)+"%");

//cimalFormat df1 = new DecimalFormat("#.##");

System.out.println("Food expenses percentage : "+df.format(foodper)+"%");

System.out.println("Logistics expenses percentage : " +df.format(logisticsper)+"%");

}

**Thrill ride**

"Fantasy Kingdom" is a brand new Amusement park that is going to be inaugurated shortly in the City and is promoted as the place for breath-taking charm. The theme park has more than 30 exhilarating and thrilling rides and as a special feature of the park, the park Authorities have placed many Booking Kiosks at the entrance which would facilitate the public to purchase their entrance tickets and ride tickets.  
   
There are few rides in the park which are not suitable for Children and aged people, hence the park Authorities wanted to program the kiosks to issue the tickets based on people’s age. If the age given is less than 15 (Children) or greater than 60 (Aged), then the system should display as "Not Allowed", otherwise it should display as "Allowed". Write a block of code to help the Authorities program this functionality.  
   
**Input Format:**  
First line of the input is an integer that corresponds to the age of the person opting for the ride.  
  
**Output Format:**  
Output should display "Allowed" or "Not Allowed" based on the conditions given.  
Refer sample input and output for formatting specifications.  
  
**Sample Input 1:**  
20  
  
**Sample Output 1:**  
Allowed  
  
**Sample Input 2:**  
12  
  
**Sample Output 2:**  
Not Allowed

Top of Form

Bottom of Form

import java.util.\*;

import java.io.\*;

class Main{

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

Scanner sc=new Scanner(System.in);

int age=sc.nextInt();

if((age<15)||(age>60)){

System.out.println("Not Allowed");

}else{

System.out.println("Allowed");}

}

}

**Character Pattern 3**

Write a program to generate a rectangular pattern of stars.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Input and Output Format:**

Input consists of a single integer that corresponds to n, the number of rows.

**Sample Input 1:**

5

**Sample Output 1:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

import java.util.\*;

import java.io.\*;

class Main{

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

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

for(int i=1;i<=n;i++){

for(int j=1;j<=i;j++){

System.out.print("\*");

}

System.out.println();

}

}

}

**Display Item Type**

The International Film Festival of India (IFFI), founded in 1952, is one of the most significant film festivals in Asia. The festival is for a weel and arrangements have to be made for food, chairs, tables, etc. The organizing committee plans to deposit the advance amount to the contractors on conformation of boking.  
Help them to store these details and print them in detailed view.  
  
Write a Java program to get item type, cost per day and deposit amount from user and display these details in a detailed view using the following classes and methods.

**[Note :  
Strictly adhere to the object oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned. Create separate classes in separate files.]**  
  
Create a class named **ItemType**with the following private member variables / attributes.

|  |  |
| --- | --- |
| **Data Type** | **Variable** |
| String | name |
| double | costPerDay |
| double | deposit |

Include appropriate **getters and setters.**  
  
In the **ItemType**class include the following methods.

|  |  |
| --- | --- |
| **Method** | **Description** |
| void display( ) | In this method, display the details of the ItemType in the format shown in the sample output. Include the statement ‘Item type details’ inside this method |

Create an another class **Main**and write a main method to test the above class.  
  
In the main( ) method, read the item type details from the user and call the display( ) method.  
  
**Example of getters and setters**  
  
private String name;  
  
public String getName( ) {  
        return name;  
}  
  
public void setName(String name) {  
        this.name = name;  
}  
  
private double costPerDay;  
  
public double getCostPerDay( ) {  
        return name;  
}  
  
public void setCostPerDay(double costPerDay) {  
        this.costPerDay = costPerDay;  
}  
  
private double deposit;  
  
public double getDeposit( ) {  
        return name;  
}  
  
public void setDeposit(double deposit) {  
        this.deposit = deposit;  
}  
**Input and Output Format:**  
Refer sample input and output for formatting specifications.  
Cost per day and Deposit value should be displayed upto 2 decimal places.  
**All text in bold correspondstoinput and the rest corresponds to output.**  
  
**Sample Input and Output 1:**  
Enter the item type name  
**Catering**  
Enter the cost per day  
**25000.00**  
Enter the deposit  
**10000.50**  
Item type details  
Name : Catering  
CostPerDay : 25000.00  
Deposit : 10000.50

Top of Form

Bottom of Form

import java.text.DecimalFormat;

public class ItemType {

private String name;

private double costPerDay,deposit;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getCostPerDay() {

return costPerDay;

}

public void setCostPerDay(double costPerDay) {

this.costPerDay = costPerDay;

}

public double getDeposit() {

return deposit;

}

public void setDeposit(double deposit) {

this.deposit = deposit;

}

public void display(){

DecimalFormat df=new DecimalFormat("0.00");

System.out.println("Item type details");

System.out.println("Name : "+getName());

System.out.println("CostPerDay : "+df.format(getCostPerDay()));

System.out.println("Deposit : "+df.format(getDeposit()));

}

}

Main

--------------

import java.io.\*;

import java.util.Scanner;

class Main{

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

ItemType i = new ItemType();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the item type name");

i.setName(sc.nextLine());

System.out.println("Enter the cost per day");

i.setCostPerDay(sc.nextDouble());

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

i.setDeposit(sc.nextDouble());

i.display();

}

}

## PROBLEM

Little App helps you discover great places to eat around or de-stress in all major cities across 20000+ merchants. Explore restaurants, spa & salons and activities to find your next fantastic deal. The development team of Little App seeks your help to find the duplication of user accounts.   
  
Write a Java program to get two users details and display whether their phone numbers are same or not with the following class and methods.  
  
**[Note : Strictly adhere to the object-oriented specifications given as a part of the problem statement.**  
**Follow the naming conventions as mentioned. Create separate classes in separate files.]**

Create a class named **User** with the following private attributes/variables.

|  |  |
| --- | --- |
| **Date Type** | **Variable** |
| String | name |
| String | username |
| String | password |
| long | phoneNo |

Include appropriate getters and setters.  
Include four-argument  constructor with parameters in the following order,  
**public User(String name, String username, String password, long phoneNo)**  
  
Include the following method in **User** class.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public boolean comparePhoneNumber(User user) | In this method, compare the phone number of the two user and return true if both the numbers are equal else return false |

Create another class **Main**and write a main method to test the above class.  
  
**Input and Output Format**  
Refer sample input and output for formatting specifications.  
**All text in bold corresponds to the input and the rest corresponds to output.  
  
Sample Input/Output 1**  
Enter Name  
**john**  
Enter UserName  
**john@123**  
Enter Password  
**john@123**  
Enter PhoneNo  
**9092314562**  
Enter Name  
**john**  
Enter UserName  
**john@12**  
Enter Password  
**john@12**  
Enter PhoneNo  
**9092314562**  
Same Users  
  
**Sample Input/Output 2**  
Enter Name  
**ram**  
Enter UserName  
**ram####**  
Enter Password  
**ram**  
Enter PhoneNo  
**9092314562**  
Enter Name  
**john**  
Enter UserName  
**john@123**  
Enter Password  
**john@123**  
Enter PhoneNo  
**9092312102**  
Different Users

Top of Form

Bottom of Form

**Rectangle Dimension Change**

Write a Java program to illustrate the method returning an objects by getting details from user and check the type of objects using instanceof and display these details in a detailed view using the following classes and methods.

**[Note :  
Strictly adhere to the object oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned. Create separate classes in separate files.]**  
  
Create a class named **Rectangle**with the following private member variables / attributes.

|  |  |
| --- | --- |
| **Data Type** | **Variable** |
| int | length |
| int | width |

Include appropriate getters and setters.

Include 2 argument constructor. The order in which the argument should be passed is**Rectangle(int length, int width)**  
  
In the **Rectangle**class include the following methods.

|  |  |
| --- | --- |
| **Method** | **Description** |
| int area( ) | This method computes the area of the rectange and returns it. |
| void display( ) | This method displays the length and width of the rectangle. Display the statement ‘Rectangle Dimension’ inside this method and also print the dimensions. |
| Rectangle dimensionChange(int d) | This method changes the rectangle dimension by increasing the length and width of the rectangle by d times. |

Create an another class **Main**and write a main() method to test the above class.  
  
In the main( ) method, read the length and width details from the user and test the above methods. Display the area of the rectange inside the main() method.  
  
**Problem Constraints:**  
1. Use **instanceof** operator to check the object returned by dimensionChange( ) method.  
**[**The java **instanceof**operator is used to test whether the object is an instance of the specified type (class or subclass or interface).**]**  
  
**Input and Output Format:**  
Refer sample input and output for formatting specifications.  
**[All text in bold correspondstoinput and the rest corresponds to output.**]  
  
**Sample Input and Output 1:**  
Enter the length of the rectangle  
**5**  
Enter the width of the rectangle  
**6**  
Rectangle Dimension  
Length:5  
Width:6  
Area of the Rectangle:30  
Enter the new dimension  
**2**  
Rectangle Dimension  
Length:10  
Width:12  
Area of the Rectangle:120

Top of Form

Bottom of Form

### Problem Requirements:

|  |  |  |
| --- | --- | --- |
| Keyword | Min Count | Max Count |
| instanceof | 1 | - |

public class Rectangle {

private int length;

private int width;

public Rectangle(int length, int width) {

super();

this.length = length;

this.width = width;

}

public int getLength() {

return length;

}

public void setLength(int length) {

this.length = length;

}

public int getWidth() {

return width;

}

public void setWidth(int width) {

this.width = width;

}

public int area() {

return (getLength()\*getWidth());

}

public void display(){

System.out.println("Rectangle Dimension");

System.out.println("Length:"+getLength());

System.out.println("Width:"+getWidth());

}

Rectangle dimensionChange(int d){

length=d\*getLength();

setLength(length);

width=d\*getWidth();

setWidth(width);

return new Rectangle(length, width);

}

}

import java.util.Scanner;

import java.io.IOException;

public class Main {

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

int length,width,area,d;

Rectangle rec;

Scanner sc=new Scanner(System.in);

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

length=sc.nextInt();

System.out.println("Enter the width of the rectangle");

width=sc.nextInt();

Rectangle r=new Rectangle(length, width);

r.display();

area=r.area();

System.out.println("Area of the Rectangle:"+area);

System.out.println("Enter the new dimension");

d=sc.nextInt();

rec=r.dimensionChange(d);

if(rec instanceof Rectangle){}

rec.display();

area=rec.area();

System.out.println("Area of the Rectangle:"+area);

}

}

**Simplified Fraction**

St. Patrick Convent organizes a project exhibition "Innovative Minds" every year with an objective to provide the platform and unleash the potential of the students by showcasing their innovative projects. Pasha is a smart high school student and was eager to participate in the fair for the first time.  
   
After a lot of ground works, she decided her project and set out to design the same. Her project requirement was to design an advanced calculator that has a fraction feature that will simplify fractions. The project will accept a non-negative integer as a numerator and a positive integer as a denominator and outputs the fraction in simplest form. That is, the fraction cannot be reduced any further, and the numerator will be less than the denominator.  
   
Help Pasha to program her advanced calculator and succeed in her first ever project presentation. You can assume that all input numerators and denominators will produce valid fractions.  
  
Hence create a class named Fraction with the following method.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| void printValue(int,int) | This method should display the fraction in simplest form. |

Create a driver class called Main. In the Main method, obtain input from the user in the console and call the printValue method present in the Fraction class.

**[Note: Strictly adhere to the Object Oriented Specifications given in the problem statement.**

**All class names, attribute names and method names should be the same as specified in the problem statement. Create separate classes in separate files.]**

**Input Format:**  
First line of the input is a non-negative integer which is the numerator in the fraction.  
Second line of the input is a positive integer which is thedenominator in the fraction.  
  
**Output Format:**  
Output the simplified form of the fraction in a single line.  
Refer sample input and output for formatting specifications.  
   
**Sample Input 1:**  
28  
7  
  
**Sample Output 1:**  
4  
  
**Sample Input 2:**  
13  
5  
  
**Sample Output 2:**  
2 3/5

Top of Form

Bottom of Form

public class Fraction {

public static void printValue (int num,int den) {

if (num <= 0) {

System.out.println(0);

}

else if (num < den) {

int gcd = getGCD(num,den);

System.out.println(num/gcd+"/"+den/gcd);

} else if (num > den) {

int rem = num % den;

int quo = (int) (num / den);

if (rem != 0) {

int gcd = getGCD(rem, den);

System.out.println(quo + " " + rem/gcd + "/" + den/gcd);

}

else {

System.out.println(quo);

}

}

else {

System.out.println(1);

}

}

private static int getGCD(int num, int den) {

int gcd = 1;

for (int i=Math.min(num, den); i>=2; i--) {

if ((num %i == 0) && (den %i == 0)) {

gcd = i;

break;

}

}

return gcd;

}

}

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int num = s.nextInt();

int den = s.nextInt();

Fraction f = new Fraction();

f.printValue(num, den);

}

}

## PROBLEM

Write a Java program to display the array of Integers and array of Strings. Use for each loop to iterate and print the elements.

**Constraints :**

Use for each loop to iterate and print the elements.

**Refer sample input and output for formatting specifications.  
All text in bold corresponds to input and the rest corresponds to output.**  
  
**Sample Input and Output 1:**

Enter n :  
**3**  
Enter numbers :   
**100  
23  
15**  
Enter strings :   
**hi  
hello  
welcome**  
Displaying numbers  
100  
23  
15  
Displaying strings  
hi  
hello  
welcome

Top of Form

Bottom of Form

**Command Line Argument - Count**

Write a program to accept strings as command line argument and print the number of arguments entered.  
  
**Sample Input (Command Line Argument) 1:**  
Command Arguments  
  
**Sample Output 1:**  
Arguments :  
Command  
Arguments  
Number of arguments is 2  
  
**Sample Input (Command Line Argument) 1:**  
Commands  
  
**Sample Output 2:**  
Arguments :  
Commands  
Number of arguments is 1

Top of Form

Bottom of Form

public class Main{

public static void main(String[] args){

int count=0;

System.out.println("Arguments :");

for(int a=0;a<args.length;a++)

{

System.out.println(args[a]);

count++;

}

System.out.println("Number of arguments is "+count);

}

}