Q1----

**DISPLAY ITEM TYPE**

**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 week and arrangements have to be made for food, chairs, tables, etc. The organizing committee plans to deposit the advance amount to the contractors on confirmation of booking.  
  
Write a Java program to get item type, cost per day, and deposit amount from the 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.]**  
  
Consider a class named **ItemType.**  
It must have the following private member variables/attributes.

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

Include the appropriate **getters and setters.**  
  
The **ItemType**class includes the following method.

|  |  |
| --- | --- |
| **Method name** | **Description** |
| public void display() | This method should display **‘Item type details’** followed by the details of the ItemType in the format as shown in the sample output. |

Consider the class **Main.**It includes the method**main**  
Write a code in the main method to test the **ItemType** class.  
The following must be done inside the main method to test the **ItemType** class.

* Get the item type details as input.
* Create an ItemType Object with the given details using the setters of ItemType and call the display( ) method**.**
* The itemType details need to be displayed in the display() method

Please use the below sample convention to create getters and setters of the class **ItemType**

private String name;

public String getName( ) {

        return name;

}

public void setName(String name) {

        this.name = name;

}

**Input and Output Format:**  
Refer sample input and output for formatting specifications.  
  
**Note:**  
**Cost per day** and **Deposit value** should be **displayed up to 2 decimal places.**  
  
**[All text in bold corresponds to input 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

**Compare Phone Number**

New 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. Write a program to find the duplication of user accounts.  
  
Write a Java program to get two users’ details and display whether their phone numbers are the 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.]**  
  
Consider the class **User**with the following private attributes/variables.

|  |  |
| --- | --- |
| **Date Type** | **Variable** |
| String | name |
| String | username |
| String | password |
| Long | phoneNumber |

Include appropriate getters and setters.  
Prototype for the Parameterized Constructor,  
**public User(String name, String username, String password, Long phoneNumber)**  
  
Define the following method in the **User**class.

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

Consider the **Main**class and write the main method to test the above class.  
  
In the main method

* Obtain the details of the user.
* Create an object for the User class using the parameterized constructor(name, username, password, phoneNumber).
* Call the method comparePhoneNumber() in the Main class.

The link to download the template code is given below  
[Code Template](https://hcl.e-box.co.in/problem/showTemplateCodeSet/18320?op=download&langid=6)  
  
**Input and Output Format**  
  
Refer sample input and output for formatting specifications.  
If both phone numbers are the same then print “**Same Users**” else print “**Different Users**”.  
The output should be printed in the **Main**class.  
  
**Sample Input/Output 1:**  
**[All text in bold corresponds to the input and the rest corresponds to output.]**  
  
Enter Name  
**john**  
Enter UserName  
**john@123**  
Enter Password  
**john@123**  
Enter PhoneNumber  
**9092314562**  
Enter Name  
**john**  
Enter UserName  
**john@12**  
Enter Password  
**john@12**  
Enter PhoneNumber  
**9092314562**  
Same Users  
  
**Sample Input/Output 2:**  
  
Enter Name  
**william**  
Enter UserName  
**william####**  
Enter Password  
**william**  
Enter PhoneNumber  
**9092314562**  
Enter Name  
**john**  
Enter UserName  
**john@123**  
Enter Password  
**john@123**  
Enter PhoneNumber  
**9092312102**  
Different Users

Top of Form

Bottom of Form

Code—

public class User {

//Fill your code

boolean flag = false;

public boolean comparePhoneNumber(User user) {

//Fill your code

return flag;

}

}

**Rectangle Dimension Change**

Write a Java program to illustrate the method of returning objects by getting details from the 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.]**  
  
Consider a class **Rectangle**with the following private member variables/attributes.

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

Include appropriate getters and setters.  
Prototype for the Parameterized Constructor,  
**public Rectangle(Integer length, Integer width)**  
  
The **Rectangle**class includes the following methods.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Integer area( ) | This method computes the area of the rectangle 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(Integer newDimension) | This method changes the rectangle dimension by increasing the length and width of the rectangle by newDimension times. |

Consider the class **Main**and write a main() method to test the above class.  
  
In the main( ) method,

* Display the area of the rectangle inside the main() method.
* Obtain the details of the user.
* Create an object for the Rectangle class using the parameterized constructor(length, width).

**Problem Constraints:**  
  
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).]  
  
The link to download the template code is given below  
[Code Template](https://hcl.e-box.co.in/problem/showTemplateCodeSet/26427?op=download&langid=6)  
  
**Input and Output Format:**  
  
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 :**  
  
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

**Array**

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 :**

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

### Problem Requirements:

#### Java

|  |  |  |
| --- | --- | --- |
| Keyword | Min Count | Max Count |
| for | 1 | 4 |

Q5—

### STRING METHODS - JAVA

**String Methods**

Write a program to implement the String methods like substring, charAt, equalsIgnoreCase, and contact.

**[ Hint: The substring begins with the character at the 3rd index and extends to the end of the first string. ]**

**Input format:**

The Input consists of two strings. String length should be greater than 5.

**Output format:**

The output consists of the result of all String methods.

**Note:**

**Refer to the sample input and output for specifications.**

**All text in bold corresponds to the input and the remaining corresponds to the output.**

**Sample Input and Output 1**

Enter the first string :   
**Amphisoft**  
Enter the second string :   
**TECHNOLOGIES**  
Substring : hisoft  
The character at 3rd position in the second string is : H  
Are string 1 and string 2 equal? : false  
Concatenated string : AmphisoftTECHNOLOGIES  
  
**Sample Input and Output 2**  
  
Enter the first string :  
**Programming**  
Enter the second string :  
**PROGRAMMING**  
Substring : gramming  
The character at 3rd position in the second string is : G  
Are string 1 and string 2 equal? : true  
Concatenated string : ProgrammingPROGRAMMING

**Command Line Argument - Count**

Write a program to accept strings as command-line arguments and print the number of arguments entered.

**Sample Input (Command Line Argument) 1:**  
Command Arguments  
  
**Sample Output 1:**  
Arguments :  
Command  
Arguments  
The number of arguments is 2

**Sample Input (Command Line Argument) 2:**  
Commands  
  
**Sample Output 2:**  
Arguments :  
Commands  
The number of arguments is 1