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

**Q.2 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 |

**Q.3**

Exercise - Dequeue

Problem Statement: Find the maximum count of unique numbers in sub array of given length

**Given an array of numbers, find the maximum count of unique numbers in sub array of given length using Dequeue**

**This exercise contains a class named UniqueNumbersCounter with the following method:**

+findUniqueNumbersCount(int[],int) : String

-Should accept int array of numbers, length of sub array as input

-Should find the maximum count of unique numbers in sub array of given length

-Should return "Give proper input not empty array" if given array is empty

-Should return "Give proper input not null array" if given array is null

-Should return "Give proper input, sub array length exceeds array length" if length of sub array is greater than length of array

-Should return "Give proper input, sub array length can not be negative or zero" if given sub array length is zero or negative

-Should return "Count of Unique Numbers is 2" if count of unique numbers in sub array of given length is 2

## **Example**

Sample Input:

array = {4,5,2,1,2}

sub array length = 3

Expected Output:

Count of Unique Numbers is 3 --------------------------------------------------------

Sample Input:

array = {6,10,4,5,2,1,2}

sub array length = 7

Expected Output:

Count of Unique Numbers is 6 --------------------------------------------------------

Sample Input:

array = {6,10,4,5,2,1,2}

sub array length = 10

Expected Output:

Give proper input, sub array length exceeds array length --------------------------------------------------------

Sample Input:

array = {7,10,2,1,2}

sub array length = -4

Expected Output:

Give proper input, sub array length can not be negative or zero

**Q.4**

**Single inheritance**

Write a Java program to implement Single Inheritance.  
  
**[Note: Strictly adhere to the object oriented specifications given as a part of the problem statement. Use the same class names and member variable names.  
Follow the naming conventions mentioned for getters/setters]**   
  
Consider a class named **Person**with the following private data members.

|  |  |
| --- | --- |
| **Data Type** | **Data Member** |
| String | name |
| String | dateOfBirth |
| String | gender |
| String | mobileNumber |
| String | bloodGroup |

Include appropriate getters and setters.  
  
Consider a class named **Donor**which extends **Person**class with the following private data members.

|  |  |
| --- | --- |
| **Data Type** | **Data Member** |
| String | bloodBankName |
| String | donorType |
| String | donationDate |

Include appropriate getters and setters.  
  
The class **Donor**should have the following method

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void displayDonationDetails( ) | This method displays the donation details. Display the statement ‘Donation Details :’ inside this method |

Consider another class **Main**and write the main method to test the above class.  
  
In the main( ) method, read the person and donor details from the user and call the displayDonationDetails( ) method.  
  
**[Note:** The date format should be **“dd-MM-yyyy”]**  
  
  
  
**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 1:**

Enter the name :  
**Justin**  
Enter Date of Birth :  
**11-01-1995**  
Enter Gender :  
**Male**  
Enter Mobile Number :  
**9994910354**  
Enter Blood Group :  
**B+ve**  
Enter Blood Bank Name :  
**Blood Assurance**  
Enter Donor Type :  
**Whole Blood**  
Enter Donation Date :  
**09-07-2017**  
Donation Details :  
Name : Justin  
Date Of Birth : 11-01-1995  
Gender : Male  
Mobile Number : 9994910354  
Blood Group : B+ve  
Blood Bank Name : Blood Assurance  
Donor Type : Whole Blood  
Donation Date : 09-07-2017  
  
  
**Sample Input and Output 2:**

Enter the name :

**Steffie**

Enter Date of Birth :

**12-01-1996**

Enter Gender :

**Female**

Enter Mobile Number :

**8868875432**

Enter Blood Group :

**O+ve**

Enter Blood Bank Name :  
**Edward Blood Bank**

Enter Donor Type :

**Whole Blood**

Enter Donation Date :

**21-12-2016**  
Donation Details :

Name : Steffie

Date Of Birth : 12-01-1996

Gender : Female

Mobile Number : 8868875432

Blood Group : O+ve

Blood Bank Name : Edward Blood Bank

Donor Type : Whole Blood

Donation Date : 21-12-2016

**Q.5**

**Abstract Class**

Write a program to calculate total cost of the event based on the type of event and display details using Abstract class and method.  
  
**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.**  
  
Consider an **abstract** class called **Event** with following protected attributes.

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| name | String |
| detail | String |
| type | String |
| organiser | String |

Prototype for the parametrized constructor, **Event(String name, String detail, String type, String organiser)**  
Include appropriate getters and setters

Include the following abstract method in the class Event.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| abstract Double calculateAmount() | an abstract method |

Consider a class named **Exhibition**which extends **Event** class with the following private attributes

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| noOfStalls | Integer |
| rentPerStall | Double |

Prototype for the parametrized constructor,  
**public Exhibition(String name, String detail, String type, String organiser, Integer noOfStalls,Double rentPerStall)**  
Include appropriate getters and setters

Use super( ) to call and assign values in base class constructor.

Include the following abstract method in the class Exhibition.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Double calculateAmount () | This method returns the product of noOfStalls and rentPerStall |

Consider a class named **StageEvent** which extends **Event**class with the following private attributes.

|  |  |
| --- | --- |
| **Attribute** | **Datatype** |
| noOfShows | Integer |
| costPerShow | Double |

Prototype for the parametrized constructor,  
**public StageEvent(String name, String detail, String type, String organiser, Integer noOfShows,Double costPerShow)**  
Include appropriate getters and setters

Use super( ) to call and assign values in base class constructor.

Include the following abstract method in the class StageEvent.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Double calculateAmount() | This method returns the product of noOfShows and costPerShow |

Consider a driver class called **Main**. In the main method, obtain input from the user and create objects accordingly.  
  
  
**Input format:**  
Input format for Exhibition is in the CSV format (**name,detail,type,organiser,noOfStalls,rentPerStall)**  
Input format for StageEvent is in the CSV format (**name,detail,type,organiser,noOfShows,costPerShow)**      
  
**Output format:**  
Print "**Invalid choice**" if the input is invalid to our application and terminate.  
Display one digit after the decimal point for Double datatype.  
Refer to sample Input and Output for formatting specifications.  
  
**[All text in bold corresponds to the input and rest corresponds to output]**  
  
**Sample Input and output 1:**

Enter your choice

1.Exhibition

2.StageEvent

**1**

Enter the details in CSV format

**Book expo,Special sale,Academics,Martin,100,1000**

Exhibition Details

Event Name:Book expo

Detail:Special sale

Type:Academics

Organiser Name:Martin

Total Cost:100000.0

**Sample Input and Output 2:**

Enter your choice

1.Exhibition

2.StageEvent

**2**

Enter the details in CSV format

**JJ magic show,Comedy magic,Entertainment,Steffania,5,1000**

Stage Event Details

Event Name:JJ magic show

Detail:Comedy magic

Type:Entertainment

Organiser Name:Steffania

Total Cost:5000.0

**Sample Input and Output 3:**

Enter your choice

1.Exhibition

2.StageEvent

**3**

Invalid choice