**Abstract Class - Introduction**

     Abstract classes are declared with a keyword abstract. These classes cannot be instantiated. They can have attributes/methods. They can have normal methods as well as abstract methods. These abstract methods must be implemented in subclasses or else they should also be declared as abstract. Now let's get on with implementing this new ly learned concept in a simple program, before going to our application.  
  
    Create an abstract class **Shape** with abstract method **public abstract Double calculatePerimeter()**  
  
    Create a class **Circle** that extends **Shape** with following attributes,

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| radius | Float |

    Create a class **Rectangle** that extends **Shape** with the following attributes ,

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| length | Float |
| breadth | Float |

    Create a class **Square** that extends **Shape** with the following attributes ,

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| side | Float |

    Implement the method **calculatePerimeter()** in all the child classes to calculate appropriate perimeters.  
  
    **Note**: Use 3.14 for pi value and display 2 digits after decimal in the answer.  
  
    Refer sample input/output for other further details and format of the output.  
  
**[All Texts in bold corresponds to the input and rest are output]  
Sample Input/Output 1:**  
  
List of Shapes:  
1.Circle  
2.Rectangle  
3.Square  
Enter your choice:  
**1**  
Enter the radius of the Circle:  
**2.34**  
The perimeter is 14.70  
  
**Sample Input/Output 2:**  
  
List of Shapes:  
1.Circle  
2.Rectangle  
3.Square  
Enter your choice:  
**2**  
Enter the length of the Rectangle:  
**12**  
Enter the breadth of the Rectangle:  
**3**  
The perimeter is 30.00  
  
**Sample Input/Output 3:**  
  
List of Shapes:  
1.Circle  
2.Rectangle  
3.Square  
Enter your choice:  
**3**  
Enter the side of the Square:  
**13**  
The perimeter is 52.00

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Assign Groups](https://app.e-box.co.in/problem/showTestcaseGroup/8047)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | s5 | 2 10 0 | List of Shapes: 1.Circle 2.Rectangle 3.Square Enter your choice: Enter the length of the Rectangle: Enter the breadth of the Rectangle: The perimeter is 20.00 | General Testcase |  | | 2 | s6 | 3 1.7 | List of Shapes: 1.Circle 2.Rectangle 3.Square Enter your choice: Enter the side of the Square: The perimeter is 6.80 | General Testcase |  | | 3 | s4 | 1 0 | List of Shapes: 1.Circle 2.Rectangle 3.Square Enter your choice: Enter the radius of the Circle: The perimeter is 0.00 | General Testcase |  | | 4 | s1 | 1 2.34 | List of Shapes: 1.Circle 2.Rectangle 3.Square Enter your choice: Enter the radius of the Circle: The perimeter is 14.70 | General Testcase |  | | 5 | s3 | 3 13 | List of Shapes: 1.Circle 2.Rectangle 3.Square Enter your choice: Enter the side of the Square: The perimeter is 52.00 | General Testcase |  | | 6 | s2 | 2 12 3 | List of Shapes: 1.Circle 2.Rectangle 3.Square Enter your choice: Enter the length of the Rectangle: Enter the breadth of the Rectangle: The perimeter is 30.00 | General Testcase |  | |

**Abstract Class - FundTransfer**

 Now we have implemented an abstract class in a simple program, we can move on to a larger program. Let's try a larger application like fund transfer before moving on to our application. So, in fundtransfer there are 3 types NEFT/IMPS/RGTS. We can create an abstract class FundTransfer. And extend it in the child classes. Create an abstract method transfer and implement in all the child classes.  
  
Create an abstract class **FundTransfer** with following attributes,

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| accountNumber | String |
| balance | Double |

and following methods,

|  |  |
| --- | --- |
| **Method** | **Description** |
| Boolean validate(Double transfer) | to check if the accountNumber is 10 digits, transfer amount is non-negative and less than balance, and return true, if not return false |
| Boolean transfer(Double transfer) | abstract method with no definition |

    Create a class **NEFTTransfer** which extends **FundTransfer** and implements transfer method,

|  |  |
| --- | --- |
| **Method** | **Description** |
| Boolean transfer(Double transfer) | check if transfer amount+5% of transfer amount is less than balance, then subtracts transfer amount and 5% service charge from balance and return true, if not return false |

    Create a class **IMPSTransfer** which extends **FundTransfer** and implements transfer method,

|  |  |
| --- | --- |
| **Method** | **Description** |
| Boolean transfer(Double transfer) | check if transfer amount+2% of transfer amount is less than balance, then subtracts transfer amount and 2% service charge from balance and return true, if not return false |

    Create a class **RTGSTransfer** which extends **FundTransfer** and implements transfer method,

|  |  |
| --- | --- |
| **Method** | **Description** |
| Boolean transfer(Double transfer) | check if transfer amount is greater than Rs.10000, then subtracts transfer amount from balance and return true, if not return false |

    Add appropriate getters/setters, constructors with super() to create objects. Write a driver class Main to test them.  
  
    **Note**: Print "Account number or transfer amount seems to be wrong" if validate function returns false. print "Transfer could not be made" if transfer function returns false.  
                Print the statements in main method.  
  
    Refer sampleinput/output for other further details and format of the output.  
  
**[All Texts in bold corresponds to the input and rest are output]  
Sample Input/Output 1:**  
  
Enter your account number:  
**1234567890**  
Enter the balance of the account:  
**10000**  
Enter the type of transfer to be made:  
1.NEFT  
2.IMPS  
3.RTGS  
**1**  
Enter the amount to be transferred:  
**2000**  
Transfer occurred successfully  
Remaining balance is 7900.0  
  
**Sample Input/Output 2:**  
  
Enter your account number:  
**1111111**  
Enter the balance of the account:  
**10000**  
Enter the type of transfer to be made:  
1.NEFT  
2.IMPS  
3.RTGS  
**2**  
Enter the amount to be transferred:  
**1000**  
Account number or transfer amount seems to be wrong  
  
**Sample Input/Output 3:**  
  
Enter your account number:  
**1234567890**  
Enter the balance of the account:  
**50000**  
Enter the type of transfer to be made:  
1.NEFT  
2.IMPS  
3.RTGS  
**3**  
Enter the amount to be transferred:  
**7500**  
Transfer could not be made

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Assign Groups](https://app.e-box.co.in/problem/showTestcaseGroup/8049)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | s3 | 1234567890 50000 3 7500 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Transfer could not be made | General Testcase |  | | 2 | s6 | 12345 100000 1 10000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Account number or transfer amount seems to be wrong | General Testcase |  | | 3 | s2 | 1111111 10000 2 1000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Account number or transfer amount seems to be wrong | General Testcase |  | | 4 | s4 | 1234567890 100000 3 10000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Transfer could not be made | General Testcase |  | | 5 | s8 | 1234567890 10500 1 10000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Transfer could not be made | General Testcase |  | | 6 | s7 | 1234567890 12000 2 11000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Transfer occurred successfully Remaining balance is 780.0 | General Testcase |  | | 7 | s1 | 1234567890 10000 1 2000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Transfer occurred successfully Remaining balance is 7900.0 | General Testcase |  | | 8 | s5 | 1234567890 1000 3 20000 | Enter your account number: Enter the balance of the account: Enter the type of transfer to be made: 1.NEFT 2.IMPS 3.RTGS Enter the amount to be transferred: Account number or transfer amount seems to be wrong | General Testcase |  | |

**Abstract Event**

Having done a few examples in abstract class you should be quite clear about abstraction and its uses. let's have a final practice in this concern. Remember creating an application for StageEvent and Exhibition from a parent class Event. Let's implement the same problem with slight changes with help of abstract class.  
  
Create an **abstract** class called **Event** with following protected attributes.

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

Event class contains the following **abstract** methods

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

Create an **Exhibition** class with following attributes

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

The Exhibition class has following methods

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

Create a **StageEvent** class with following attributes.

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

The StageEvent class has the following methods

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

Include appropriate getters and setters for all the classes.  
create a default constructor and parameterized constructor for all the classes.  
  
The format for Parameterized constructors:  
**public Event(String name, String detail, String type, String organiserName)**  
**public Exhibition(String name, String detail, String type,String organiser,Integer noOfStalls,Double rentPerStall)**  
**public StageEvent(String name, String detail, String type,String organiser,Integer noOfShows,Double costPerShow)**  
  
Input format for Exhibition input: **name,detail,type,organiser,noOfStalls,rentPerStall**  
Input format for StageEvent input: **name,detail,type,organiser,noOfShows,costPerShow**      
  
Create a driver class called **Main**. In the Main method, obtain input from the user and create objects accordingly.  
Note: Print "Invalid choice" if the input is invalid and terminate.  
display one digit after decimal point for Double datatype  
**[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.]**  
  
**[All text in bold corresponds to the input and rest corresponds to output]**  
**Sample Input/output 1:**

Enter your choice

1.Exhibition

2.StageEvent

**1**

Enter the details in CSV format

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

Exhibition Details

Event Name:Book expo

Detail:Special sale

Type:Academics

Organiser Name:Mahesh

Total Cost:100000.0

**Sample Input/Output 2:**

Enter your choice

1.Exhibition

2.StageEvent

**2**

Enter the details in CSV format

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

Stage Event Details

Event Name:JJ magic show

Detail:Comedy magic

Type:Entertainment

Organiser Name:Jegadeesh

Total Cost:5000.0

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Assign Groups](https://app.e-box.co.in/problem/showTestcaseGroup/8051)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | s2 | 2 JJ magic show,Comedy magic,Entertainment,Jegadeesh,5,1000 | Enter your choice 1.Exhibition 2.StageEvent Enter the details in CSV format Stage Event Details Event Name:JJ magic show Detail:Comedy magic Type:Entertainment Organiser Name:Jegadeesh Total Cost:5000.0 | General Testcase |  | | 2 | s6 | 2 Food fest, Cooking,Buffet,Bilal,1,20000 | Enter your choice 1.Exhibition 2.StageEvent Enter the details in CSV format Stage Event Details Event Name:Food fest Detail: Cooking Type:Buffet Organiser Name:Bilal Total Cost:20000.0 | General Testcase |  | | 3 | s1 | 1 Book expo,Special sale,Academics,Mahesh,100,1000 | Enter your choice 1.Exhibition 2.StageEvent Enter the details in CSV format Exhibition Details Event Name:Book expo Detail:Special sale Type:Academics Organiser Name:Mahesh Total Cost:100000.0 | General Testcase |  | | 4 | s4 | 7 | Enter your choice 1.Exhibition 2.StageEvent Invalid choice | General Testcase |  | | 5 | s5 | 1 Car expo,Imported cars,Auction,Ramkumar,20,100000 | Enter your choice 1.Exhibition 2.StageEvent Enter the details in CSV format Exhibition Details Event Name:Car expo Detail:Imported cars Type:Auction Organiser Name:Ramkumar Total Cost:2000000.0 | General Testcase |  | | 6 | s3 | 3 | Enter your choice 1.Exhibition 2.StageEvent Invalid choice | General Testcase |  | |