Hands-on Assignments for Abstract Classes

No. Hands-on Assignment Topics Covered Status

1

1.1. Create a class called GeneralBank which acts as base class for all banks. This class has functionality getSavingInterestRate and getFixedInterestRate methods, which return the saving a/c rate of interest and fixed account rate of interest the specific bank gives. Since GeneralBank cannot say what percentage which bank would give, make it abstract.

1.2. Create 2 subclasses of GeneralBank called ICICIBank and KotMBank. Override the methods from base class. ICICI - Savings 4% Fixed 8.5% and KotMBank. - Savings 6% Fixed 9%

1.3. Create a main method to test the above classes. Try one by one and absorb your finding. a) ICICIBank object reference instantiated with ICICIBank class. b) KotMBank object reference instantiated with KotMBank class. c) GeneralBank object reference instantiated with KotMBank class. d) GeneralBank object reference instantiated with ICICIBank class.

Abstract Classes

2

Create an abstract class Compartment to represent a rail coach. Provide an abstract function notice in this class. Derive FirstClass, Ladies, General, Luggage classes from the compartment class. Override the notice function in each of them to print notice suitable to the type of the compartment. Create a class TestCompartment . Write main function to do the following: Declare an array of Compartment of size 10. Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4. Check the polymorphic behavior of the notice method.

Abstract Classes

3

Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message “Piano is playing tan tan tan tan ” for Piano class “Flute is playing toot toot toot toot” for Flute class “Guitar is playing tin tin tin ” for Guitar class You must not allow the user to declare an object of Instrument class. Create an array of 10 Instruments. Assign different type of instrument to Instrument reference. Check for the polymorphic behavior of play method. Use the instanceof operator to print that which object stored at which index of instrument array.

Abstract Classes

Hands-on Assignments for Interfaces

No. Hands-on Assignment Topics Covered Status

1

A library needs to develop an online application for two types of users/roles, Adults and children. Both of these users should be able to register an account.

Any user who is less than 12 years of age will be registered as a child and they can borrow a "Kids" category book for 10 days, whereas an adult can borrow "Fiction" category books which need to be returned within 7 days.

Note: In future, more users/roles might be added to the library where similar rules will be enforced.

Develop Interfaces and classes for the categories mentioned above.

1. Create an interface LibraryUser with the following methods declared, Method Name registerAccount requestBook
2. Create 2 classes "KidUsers" and "AdultUser" which implements the LibraryUser interface.
3. Both the classes should have two instance variables as specified below. Instance variables Data type age int bookType String
4. The methods in the KidUser class should perform the following logic. registerAccount function: if age < 12, a message displaying "You have successfully registered under a Kids Account" should be displayed in the console. If(age>12), a message displaying, "Sorry, Age must be less than 12 to register as a kid" should be displayed in the console. requestBook function: if bookType is "Kids", a message displaying "Book Issued successfully, please return the book within 10 days" should be displayed in the console. Else, a message displaying, "Oops, you are allowed to take only kids books" should be displayed in the console.
5. The methods in the AdultUser class should perform the following logic.

registerAccount function: if age > 12, a message displaying "You have successfully registered under an Adult Account" should be displayed in the console. If age<12, a message displaying, "Sorry, Age must be greater than 12 to register as an adult" should be displayed in the console. requestBook function: if bookType is "Fiction", a message displaying "Book Issued successfully, please return the book within 7 days" should be displayed in the console. Else, a message displaying, "Oops, you are allowed to take only adult Fiction books" should be displayed in the console. 6. Create a class "LibraryInterfaceDemo.java" with a main method which performs the below functions,

Test case #1: Create an instance of KidUser class. Set the age as specified in the below table and invoke the registerAccount method of the KidUser object

Age 10 18

Set the book Type as specified in the below table and invoke the requestBook method of the KidUser object,

BookType “Kids” “Fiction”

Test case #2:

Create an instance of AdultUser class. Set the age as specified in the below table and invoke the registerAccount method of the AdultUser object

Age 5 23

Set the book Type as specified in the below table and invoke the requestBook method of the AdultUser object BookType “Kids” “Fiction”

Interfaces

2

Write an interface called Playable, with a method void play(); Let this interface be placed in a package called music.

Write a class called Veena which implements Playable interface. Let this class be placed in a package music.string

Write a class called Saxophone which implements Playable interface. Let this class be placed in a package music.wind

Write another class Test in a package called live. Then, a. Create an instance of Veena and call play() method b. Create an instance of Saxophone and call play() method c. Place the above instances in a variable of type Playable and then call play()

Interfaces