

Java for Non Majors

CGS 3416: Spring 2016

Homework 7: 200 points

Due date: 11:59 PM 04/12/2016

Objective

The purpose of this assignment is to test your proficiency with Java Classes. It will also serve to check if you can integrate older concepts with newer ones.

For this assignment you are required to write two programs. Please email your files "`Books.java`" and "`TestShapes.java`" to jayarama@cs.fsu.edu

Problem - A Mini Library

You find yourself inexplicably in the Boston Public Library in Fallout 4. Your task is to return "Daisy's book". You have no way of knowing which one it is. It could be any of the five million books strewn around. You, being a rather intelligent Java programmer decide to hack into the intercom system to print basic information about every book it finds, so that you can quickly locate Daisy's book.

Specifications

- This program is worth 100 points and should be called `Books.java`
- Create a class with the appropriate name. (5 points)
- The class should be in a package called "`Library`", and should be declared public.(5 points)
- The class has three data attributes - `title`, `donatedBy` and `numChapters`. (5 points)
- Write accessor and mutator methods for each of the attributes. Please follow Java naming convention for these methods. (10 points)
- Write a default constructor that sets the value of the book to `{Java Programming, Kathy Sierra, 21}` (5 points)
- Write a parametrized constructor. If the given number of chapters is 0 or negative, it should be set to the default value of 21. (10 points)
- Write a method called `printDetails` that prints all the details of a book. (10 points)
- The main method should be the only static method in the class.
- In the main method, accept the number of books 'N' from the user. Then create an array of 'N' objects of the Books class. Read in the values of the books from the user and use the constructors to set the values of the objects. Then, call the `printDetails` method to print the details of the books, one by one. If any of the books was donated by "Daisy", print "Task Complete." at the very end. (45 points)
- Please include comments wherever appropriate. (5 points)

Compatibility

Once your class is compiled into a classfile, I should be able to instantiate (create objects) the class in another program and use the objects. For example, I should be able to compile the Books class inside the Library directory, and then run the ManyBooks class from the parent directory. The ManyBooks class can be found at <http://ww2.cs.fsu.edu/~jayarama/java16/Examples/ManyBooks.java>

Sample Run

Regular text is what's printed by your program. Underlined text is user input, shown here as a sample. You will not be printing the underlined parts in your program.

```
Enter the number of books: 3
Enter the title: Concrete Mathematics
Enter the name of the donor: Tim Barao
Enter the number of chapters: 12
Enter the title: Auto Repair for Dummies
Enter the name of the donor: Daisy
Enter the number of chapters: 16
Enter the title: Is your Cat Trying to Kill You?
Enter the name of the donor: Jane Smith
Enter the number of chapters: 35
```

```
Book 1:
Title: Concrete Mathematics
Donor: Tim Barao
Chapters: 12
```

```
Book 2:
Title: Auto Repair for Dummies
Donor: Daisy
Chapters: 16
```

```
Book 3: Title: Is your Cat Trying to Kill You?
Donor: Jane Smith
Chapters: 35
```

Task Complete.

Problem 2 - What's that shape?

You have been commissioned to write a program to analyze given shapes and print their color and area. To save a lot of time and effort, you decide on an hierarchical structure. You figure all shapes have certain things in common. So, you write a common Shapes class to contain those details and then extend the other shape-specific classes from the Shape class. Since Java is pretty great for this setup, you decide to write a Java Console application to do this for you.

Specifications

This program should be called `TestShapes.java` and is worth 100 points. For this program you are required to write 5 classes. The requirements are listed below. Please follow the given naming guidelines.

- Define a class called **Shape**. (15 points)
 - This class just has one attribute - `color`.
 - Write a default constructor that sets color to “red”.
 - Write a parametrized constructor and accessor and mutator methods.
 - Write a method called `print`, that takes no parameters and prints the color.
 - Also define a method called “`area`” that returns double, but leave it empty.
- Define a class called **Square**. (15 points)
 - This class inherits from the **Shape** class.
 - The class has the attribute `sideLength` - `double`.
 - Write a default constructor that sets `sideLength` to 1.

- Write the parametrized constructor, accessor and mutator methods. Make sure to invoke the superclass' constructor appropriately in both of the constructors.
- Override the print and area methods. In the print method, call the superclass' print method as well. The print method should then print the sideLength and the area of the square. In the area method, just calculate the area and return it.
- Define a class called **Rectangle** (15 points)
 - This class inherits from the **Shape** class.
 - The class has two attributes **length - double** and **width - double**.
 - Write a default constructor that sets length and width to 1.
 - Write the parametrized constructor, accessor and mutator methods. Make sure to invoke the superclass' constructor appropriately in both of the constructors.
 - Override the print and area methods. In the print method, call the superclass' print method as well. The print method should then print the length and width, and the area of the rectangle. In the area method, just calculate the area and return it.
- Define a class called **Circle** (15 points)
 - This class inherits from the **Shape** class.
 - The class has the attribute **radius - double**.
 - Write a default constructor that sets radius to 1.
 - Write the parametrized constructor, accessor and mutator methods. Make sure to invoke the superclass' constructor appropriately in both of the constructors.
 - Override the print and area methods. In the print method, call the superclass' print method as well. The print method should then print the radius and the area of the circle. In the area method, just calculate the area and return it.
- Define a class called **TestShapes**.
 - This class should only contain the main method.
 - Accept a number 'N' from the user. Then create an array of type **Shape** of size N. (5 points)
 - Ask the user to choose between the 3 shapes and enter an integer to denote their choice. If the user enters 1, it is a square. If the user enters 2, it is a rectangle, and if the user enters 3, it is a circle. You may assume that the user will only enter 1, 2 or 3. (10 points)
 - Read in the required attributes (color, and whatever is needed for the user's choice. Create an object of the appropriate class according to the user's choice and attach it to the reference in the array. (10 points)
 - Once the array of objects is created, invoke the print method for each object in the array one by one. (10 points)
- Please include comments wherever appropriate. (5 points)

Sample Run

Regular text is what's printed by your program. Underlined text is user input, shown here as a sample. You will not be printing the underlined parts in your program.

```
Enter the number of shapes: 3
Enter the choice (Square, Rectangle or Circle): 1
Enter the color: Green
Enter the side length of the square: 12
Enter the choice (Square, Rectangle or Circle): 3
Enter the color: Purple
Enter the radius of the circle: 3
Enter the choice (Square, Rectangle or Circle): 2
Enter the color: Blue
Enter the length of the rectangle: 5
```

Enter the width of the rectangle: 9

Shape 1:
Color: Green
Side Length: 12
Area: 144

Shape 2:
Color: Purple
Radius: 3
Area: 28.27

Shape 3:
Color: Blue
Length: 5
Width: 9
Area: 45

Generic Guidelines

- Please add your name and FSUID as comments on the top of your program.
- Please make sure you're only using the concepts already discussed in class. These assignments are used to determine if you passed a certain learning milestone. So, please follow the specifications. Using concepts that are not in the specifications will result in a score of 0 points for that particular specification.
- Please make sure that you're conforming to output specifications (program name, expected inputs and outputs etc.). Your output must match the sample output exactly (especially the literal text on print statements).
- Please make sure your code is readable and well documented.
- Make sure to compile and run your program before you turn it in. Compilation errors can be costly.
- You can use an IDE for this assignment, but make sure your program compiles and runs on a terminal, since the program will be tested on a terminal.