CS 46A Fall 2022

Homework 03

Requirements

1. You must name your classes **exactly** as specified. Otherwise Codecheck will not be able to process your submission and you will get no credit.
2. When you are finished with your code, submit it to Codecheck one final time then download the .signed.zip file.
3. You must upload all three signed.zip files together to Canvas and you should double check the files in Canvas to make sure all three zip files are uploaded.
4. Do not open the downloaded zip files. The files are digitally signed, and the grader program will check that they have not been opened.
5. Due time: 10 pm, Saturday, Sep 17. Submissions before the due time are not late.

   You will lose five points if your submission is marked Late in Canvas.

6. Grace time: 10 am, Sunday, Sep 18. Submissions before the grace time will not be rejected.

   You will receive no points if your submission is rejected by Canvas.

Remember to follow our Programming Style Requirements!

Problem 3A

Create a new BlueJ project with two classes, Climber and ClimberTester, then copy both classes from Codecheck. Do not change class ClimberTester in any way.

Complete the class Climber which represents a child's position on a climbing pole. The position can be anywhere from the bottom to the top of the pole. The pole is 10 unit high. At any point, the child can slide back to the bottom of the pole. We want to be able to have the child climb and slide. We want to be able to get his name and his current position.

What will the object need to remember? The child’s name and his position on the pole - those are the instance variables.

The class has one constructor :

- **`public Climber (String theName, int thePosition)`** Constructs a new climber with the specified name and position. Remember that the job of the constructor is to initialize the instance variables.

The class has these methods:

- `public String getName()` Gets the name of this Climber object.
- `public int getPosition()` Gets the position on the pole of this Climber object.
- `public void setName(String newName)` Sets a new name for this Climber object.
- `public void climb()` Climbs one unit on the pole (add one to this climber position to simulate climbing one position on the pole).
- `public void slide()` Sets the climber back to the bottom of the pole.

Javadoc is provided for this class. Study how it is done so you can write the required Javadoc in the next problem.

Codecheck link for 3A

Problem 3B

HAL 9000 is a fictional character in Arthur C. Clarke's 2001: A Space Odyssey. HAL (Heuristically programmed ALgorithmic computer) is an AI (artificial intelligence) computer that controls the system of the spacecraft and interacts with the ship's crew.

Create a BlueJ project with two classes, Hal9000 and Hal9000Tester, then copy the tester class from Codecheck. Do not change the tester class in any way.

You will write the entire `Hal9000` class and also supply Javadoc for the class.

The class has one constructor that takes a String parameter for the name of the crew member it is interacting with.

The class has the following methods:

- `public String getName()` Gets the name of the crew member.
- `public void setName(String newName)` Sets a new name.
- `public String greet() R`eturns a string "Greetings, Dave." if the crew name is Dave.
- `public String giveStatus()` Returns a string "Everything is a go, Dave.” if the crew name is Dave.
- `public String executeCommand(String whatToDo)` Returns a string "I am sorry, Dave. I can't engage drive." if the crew name is Dave and the command (the parameter whatToDo) is “engage drive”.

Codecheck link for 3B

Problem 3C

In this exercise, we are going to revisit the snowman problem of homework02. In that assignment you wrote a **`SnowmanViewer`** with a main() method and did all the drawing in it. But there is a problem with that design: If you wanted to draw a second snowman (or a third or a fourth), you would have to recalculate all the coordinates and repeat all the lines of code. Not good! The solution to this problem is to create a **`SnowMan`** class that models a snowman at a given location. Then that class can be used in different applications to draw SnowMan objects at any locations.

Create a BlueJ project and import the graphics package, then create a new class **`SnowMan`**. There is no starter code for the class.

The class has two instance variables to store the specified position. It has a constructor and one method draw(). The constructor takes the x and y coordinates of the upper-left corner of the rectangular hat of the snowman as parameters. Do not draw in the constructor.

In the draw() method, draw a rectangle of size 20 by 20 as the hat at the specified location (x, y), then fill it using the default color BLACK. After that, draw the hat brim and three circles to make it a snowman as you did for 2C. You need to figure out the coordinates for each object, but they all have the same size as in 2C. Copy and paste the following static constants to your code. Do not have any "magic" numbers! For this problem, number 2 is not considered as a magic number.

```
public static final int HAT_SIZE = 20;
public static final int HAT_BRIM_SIDE_SIZE = 10;
public static final int SMALL_DIAMETER = 20;
public static final int DIAMETER_INCREMENT = 20;
public static final int RADIUS_INCREMENT = 10;
```

Create another class **`SnowManProg`** and copy the code from Codecheck to test your class. Submit only your class **`SnowMan`** to Codecheck.

Codecheck link for 3C