

Programming Assignment 2 (Eclipse)

Due Sep 21, 2019 by 11:59pm **Points** 30 **Submitting** a website url or a file upload
File Types zip

Programming Assignment 2 will consist of writing multiple short programs. You will write these programs from scratch, using the concepts covered so far (variables, operators, selection statements, loops, and arrays).

Each problem below should be self contained within its own Eclipse project. You will upload these projects to your GitHub site. Each project must contain the java source code, the project files created by Eclipse, *and* the .class files generated when you compile your program. To include the .class files, you may need to modify your .gitignore file. All three problems should be in the *same* repository. To keep things simple, I suggest that you create a new repository on GitHub called **<last_name>_PA2** and push your projects there (instead of the same repo you used for Assignment 1).

You will submit your assignment as *both* a link to your GitHub page and an upload of your projects as a *.zip file. Your **projects should be named <last_name>_pN**, where N is the problem number the corresponds to each project. You must **name your zip file <last_name>_PA2.zip**. Failure to adhere to these naming convention may result in your assignment going ungraded. Due to Webcourse@UCF limitations, when you submit your assignment, submit the zip file. Add your GitHub URL as a note during the submission, or as a comment after the submission.

An example structure might be:

```
hollander_PA2.zip
|--- hollander_p1
|--- hollander_p2
|--- hollander_p3
```

Grades for this program will be determined according to the rubric. **If your project on GitHub does not contain the complete Eclipse project structure (i.e. the .project file, .classpath file, your .java files, and your .class files), it will not be graded.** This requirement is in place so that my graders can either pull your code directly from git or unzip it and run it without modification. Similarly, if your project does not run when we try and open it, it will not be graded. This is most likely to occur if you put all of your files in the same directory. By default, Eclipse separates the source and binary files; so your GitHub repo should reflect this structure (it's defined in your .classpath file).



Problem 1

A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely. All the data is transmitted as four-digit integers. Your application should read a four-digit integer entered by the user and encrypt it as follows: Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10. Then swap the first digit with the third, and swap the second digit with the fourth. Then print the encrypted integer. Write a separate application that inputs an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number.



Problem 2

The formulas for calculating BMI are

$$BMI = \frac{703 \cdot \text{weightInPounds}}{\text{heightInInches}^2}$$

or

$$BMI = \frac{\text{weightInKilograms}}{\text{heightInMeters}^2}$$

Create a BMI calculator that reads the user's weight and height (providing an option for the user to select which formula to use), and then calculates and displays the user's body mass index. Also, display the BMI categories and their values from the National Heart Lung and Blood

Institute: http://www.nhlbi.nih.gov/health/educational/lose_wt/BMI/bmicalc.htm

http://www.nhlbi.nih.gov/health/educational/lose_wt/BMI/bmicalc.htm so the user can evaluate his/her BMI.



Problem 3

Write a simple polling program that allows users to rate five topics from 1 (least important) to 10 (most important). Pick five topics that are important to you (e.g., political issues, global environmental issues, food, video games). Use a one-dimensional array `topics` (of type `String`) to store the five issues. To summarize the survey responses, use a 5-row, 10-column two-dimensional array `responses` (of type `int`), each row corresponding to an element in the `topics` array. When the program runs, it should ask the user to rate each issue. Multiple people should be able to respond to the survey during a single

run of the program. Once all responses have been logged, have the program display a summary of the results, including:

1. A tabular report with the five topics down the left side and the 10 ratings across the top, listing in each column the number of ratings received for each topic.
2. To the right of each row, show the average of the ratings for that issue.
3. Which issue received the highest point total? Display both the issue and the point total.
4. Which issue received the lowest point total? Display both the issue and the point total.

Programming Assignment 2

Criteria	Ratings		Pts
p1: main method	1 pts Full Marks	0 pts No Marks	1 pts
p1: prompt for number	1 pts Full Marks	0 pts No Marks	1 pts
p1: get number from user	1 pts Full Marks	0 pts No Marks	1 pts
p1: display correctly encrypted value	1 pts Full Marks	0 pts No Marks	1 pts
p1: display correctly decrypted value	1 pts Full Marks	0 pts No Marks	1 pts
p2: main method	1 pts Full Marks	0 pts No Marks	1 pts
p2: prompt for calculation mode (lb or kg)	1 pts Full Marks	0 pts No Marks	1 pts
p2: prompt for weight and height input	1 pts Full Marks	0 pts No Marks	1 pts
p2: get calculation mode from user	1 pts Full Marks	0 pts No Marks	1 pts
p2: get weight from user	1 pts Full Marks	0 pts No Marks	1 pts
p2: get height from user	1 pts Full Marks	0 pts No Marks	1 pts

Criteria	Ratings		Pts
p2: correct BMI calculation	1 pts Full Marks	0 pts No Marks	1 pts
p2: display of BMI value	1 pts Full Marks	0 pts No Marks	1 pts
p2: display of BMI category	1 pts Full Marks	0 pts No Marks	1 pts
p3: prompt user to rate each topic	1 pts Full Marks	0 pts No Marks	1 pts
p3: get topic ratings from user	1 pts Full Marks	0 pts No Marks	1 pts
p3: user input phase occurs within a loop	1 pts Full Marks	0 pts No Marks	1 pts
p3: display tabular report as a 5x10 table	1 pts Full Marks	0 pts No Marks	1 pts
p3: include per-topic average in report	1 pts Full Marks	0 pts No Marks	1 pts
p3: display issue with highest point total	1 pts Full Marks	0 pts No Marks	1 pts
p3: display issue with lowest point total	1 pts Full Marks	0 pts No Marks	1 pts
p1: project on GitHub	3 pts Full Marks	0 pts No Marks	3 pts

Criteria	Ratings		Pts
p2: project on GitHub	3 pts Full Marks	0 pts No Marks	3 pts
p3: project on GitHub	3 pts Full Marks	0 pts No Marks	3 pts
Total Points: 30			