Weekend Homework (400 points)

**How to submit à Submit 4 files to your GitHub repository. Then, send your GitHub repository link via email and provide the link in this assignment dropbox (no file but provide link in the textbox). Due is Tuesday, August 30, 2022**

**(1) Submit this file and change the file name to WeekendHomeworkYOUR\_FULLNAME.docx. (100 points)**

**(2) Submit Escaping.java (100 points)**

**(3) Submit TwoRocket9s.java (100 points)**

**(4) Submit a program you write regarding the coding standard section question. (100 points)**

1. Convert each of the following decimal numbers into its equivalent binary number:
   1. 6 🡪 110
   2. 44 🡪 101100
   3. 72 🡪 1001000
   4. 131 🡪 10000011
   5. -6 🡪-110
   6. -44 🡪-101100
   7. -72 🡪-1001000
   8. -131 🡪-10000011
2. What is the decimal equivalent of each of the following binary numbers?
   1. 100 🡪 4
   2. 1011 🡪11
   3. 101010 🡪42
   4. 1001110 🡪 78
3. Which of the following can be used in a Java program as identifiers? Check all of the identifiers that are legal.
   1. ABC
   2. "hello"
   3. for
   4. 42isThesolution
   5. AnnualSalary
   6. B4
   7. println
   8. \_average
   9. sum\_of\_data
   10. first-name
4. Which of the following is the correct syntax to output a message?
   1. System.println(Hello, world!);
   2. System.println.out('Hello, world!');
   3. System.println("Hello, world!");
   4. System.out.println("Hello, world!");
   5. Out.system.println"(Hello, world!)";
5. Write a complete Java program in a class named Escaping that prints the following output (Line one: 22 forward slashes, Line two: ||||| YOUR NAME |||||, Line three must have 22 backward slashes – read this. <https://riptutorial.com/java/example/24928/matching-a-backslash>. 🡪 Write the program in your GitHub repository. You can test with IntelliJ to check if you write the program correctly. But you must submit your code to your GitHub repository.

//////////////////////

|||||| YOUR NAME |||||

\\\\\\\\\\\\\\\\\\\\\\

1. Write a complete Java program in a class named TwoRockets that generates the following output. Use static methods to show structure and eliminate redundancy in your solution. 🡪 Write the program in your GitHub repository. You can test with IntelliJ to check if you write the program correctly. But you must submit your code to your GitHub repository.

System.out.println(" /\\ /\\ ");

System.out.println(" /\\ /\\ ");

System.out.println(" /\\ /\\");A picture containing text, antenna

Description automatically generated

1. Go to <http://itec2140.gitlab.io/codes/CodingStandards.pdf> and read the coding standards. Create your own class and describe each section how you applied each standard.

//Add your program here.

//then, provide the evidence for each requirement. Your answers go after 🡪

**1. Variable names should be descriptive. Simple variables such as a, t or x should be avoided.  
Variable names should begin with a lowercase letter and the first letter of each word should be  
capitalized. For example, firstPerson. The use of fp here would not be a descriptive variable  
name. 🡪**

**2. Class names should begin with an uppercase letter and the first letter of each word should be  
capitalized. For example, GoodClassName. 🡪**

**3. Constant names should be all upper case. Words should be separated by and underscore. For  
example, CONSTANT\_NUMBER.** 🡪

**4. Indentation should provide the reader with a context for their location. Brackets should start on  
the same line and the start of this line should line up with the matching closing bracket.** 🡪

**5. Javadoc comments should be used to describe classes and methods.  
a. Example class comment  
/\*\*Class: ClassName  
\* @author Your Name  
\* @version 1.0  
\* Course: ITEC 2140 Fall 2022  
\* Written: August 27, 2022  
\*  
\* This class – now describe what the class does  
\*/  
• ClassName should be the name of the class. It should not be  
ITEC 2140, ITEC 2150 or some course number. This is used to  
identify you class. If it is a public class, it will be the  
file name without the .java extension.  
• Author – You must include your name in your program. Programs  
submitted without a name will receive a deduction  
• Version – should start 1.0. For minor changes should receive  
a point upgrade, 1.1 etc. If a second submission is required,  
this should be updated to 2.0 etc.  
• Course – what course and semester is this program for.  
• Written – what date did you create/modify this program.  
• Class Description – describe what the class should do at a  
high level.**

**b. Example Method comment  
/\*\* Method: Method Name  
\* Convert calendar date into Julian day.  
\* Note: This algorithm is from Press et al., Numerical Recipes  
\* in C, 2nd ed., Cambridge University Press, 1992  
\* @param day day of the date to be converted  
\* @param month month of the date to be converted  
\* @param year year of the date to be converted  
\* @return the Julian day number that begins at noon of the  
\* given calendar date.  
\*/  
 .**

**6. Use blank space to improve readability. For example, when writing an expression with binary  
operators, use a blank around the operators.  
Good – (-b + 3)  
Not good – (-b+3)** 🡪

**7. Go back your two programs Escaping.java and TwoRockets.java. Then, make sure to revise the code based on this coding standard. If you do not follow this standard. If each program does not follow the coding standard, then 10 points will be deducted.**