COM S/SE 319 : Software Construction and User Interfaces
Fall 2018

HW 4

[Total Points: 50]

Assignment Due: Sunday, October 7, 2018, 11:59 PM

[N.B.:5% penalty per day up to maximum of 7 days after **October 7, 2018**]

This assignment is focused on node.js

Task 1: (45 points)

Objectives:

Learn to use node.js programming.

Warm-up:

NOTE 1: Play with the given example. Open using a text editor of your choice and modify to learn how the different instructions work.

Task:

*It will be a console based application:

Your assignment is to **create a simple binary calculator programs**. This calculator should look approximately like the given warm-up exercise.

For **binary calculator**,

1. Note that for some operations on the binary calculator, it may be more convenient to convert the binary numbers to integers and then do the operation. (It is a suggestion, you can implement your own logic).
2. You can assume that only positive binary numbers are represented and used. For example, positive 9 is represented as 1001.
3. Binary operator "+" represents plus operation **(5 points)**
4. Binary operator "*" represents multiply **(5 points)**
5. Binary operator "/" represents division **(5 points)**
6. Binary operator "%" represents mod or remainder (i.e. divide the first value by the second, what is remaining, only works on positive numbers) **(5 points)**
7. Unary operator "<<" represents one bit-shift left (i.e. insert zeros in the vacated position on the left, only works on positive numbers) e.g. (101 << gives 1010) **(5 points)**
8. Unary operator ">>" represents one bit-Shift right (insert zeros in the vacated position on the right, only works on positive numbers) e.g. (101 >> gives 10) **(5 points)**
9. Binary operator "&" represents AND (only works on positive numbers) e.g. (101 & 1011 gives 0001) **(5 points)**

10. Binary operator "|" represents OR (only works on positive numbers) e.g. (101 | 1010 gives 1111) (**5 points**)
11. Unary operator "~" represents not (i.e. invert each bit of the binary value, only works on positive numbers) e.g. (101 ~ gives 10) (**5 points)**

What to Submit:

Submit via Canvas a **compressed file (.zip)** containing the following:

- ***code(s)*** for **Task 1**. **[Task 1= 45 Points]**
- README file explaining how to compile and run your program & a **Report** (.docx or .pdf) describing your solution approach and **screenshots** of every required output. **[5 points]**.