FALL 2022 – CS203 Object Oriented Programming
Homework 1

Deadline: 09/04/2022 Sunday 11:59pm

Objectives:

- JAVA Practice

Create the necessary JAVA files/functions to solve the problems below. Feel free to explore the JAVA syntax online, we suggest you use java official documentation.

HW1 Questions

`cubeOfOdd (n1)`

Write the function "**`cubeOfOdd`**" that takes an int **`n1`** as input. The function will print **i^3** value for *all non-negative odd integers* **`i<n1`**.

Sample Input:	**Expected Output:**	***Hints***
n1 = 5	1 27	$1^3=1$ $3^3=27$
n1 =3	1	$1^3=1$
n1 =8	1 27 125 343	$1^3=1$ $3^3=27$ $5^3=125$ $7^3=343$

introToJava (n2)

Write the function `introToJava` that takes a positive integer **n2** and prints a string according to the following conditions

- if n2 is divisible by 7, it should print "UAB"
- if n2 is divisible by 3, it should print "CS"
- if n2 is divisible by both 3 and 7, it should print "UAB CS 203"
- if n2 is a prime number other than 3 or 7, it should print "Go Blazers"
- otherwise, it should print the cube of n2

Sample Inputs	Expected Outputs
n2=3	"CS"
n2=70	"UAB"
n2=4	64
n2=17	"Go Blazers"

printJAVA (n3)

Write the function `printJAVA` that takes an integer **n3** (n is equal or greater than 0) and returns a string with the integers 0 through n3, inclusive, except that every power of 2 is replaced by *JAVA*.

Sample Inputs	Expected Outputs
n3=3	0JAVAJAVA3
n3=7	0JAVAJAVA3JAVA567
n3=10	0JAVAJAVA3JAVA567JAVA910
n3=1	0JAVA

* Note that there is no whitespace in the string.

* Two hints for a simpler, cleaner implementation: you actually do not need to do any type of division (float division, integer division, or modulo arithmetic using %) or exponentiation.
* Context: the powers of 2 are naturally of fundamental importance to computer science, since computers store integers as binary numbers (base 2). An understanding of the function 2^n is important for an appreciation of exponential-time algorithms (and its associated exponential explosion).

numSteps (n)

Write a function "**numSteps**" that takes a positive int **"n"** and returns the minimum number of steps to reduce it to zero. Here are the rules;

a) If the current number is even, you have to divide it by two
b) If the current number is odd, you need to subtract 1 from it

Sample Input

```
14
```

Sample Output

```
6
```

Hints

Step 1) 14 is even; divide by 2 and obtain 7. Step 2) 7 is odd; subtract 1 and obtain 6. Step 3) 6 is even; divide by 2 and obtain 3. Step 4) 3 is odd; subtract 1 and obtain 2. Step 5) 2 is even; divide by 2 and obtain 1. Step 6) 1 is odd; subtract 1 and obtain 0.

grader (avg_exams, avg_hw, attendance)

Write the function "**grader**" that takes two floats; **avg_exams** and **avg_hw**, and one integer **attendance** as input and returns a **bool** value. The function will check any student's grade and attendance to decide if the student pass the course or fail. If the student pass, the function prints "PASS", otherwise it prints "FAIL". These are the criteria to pass the course;

- The attendance must be greater than 20 to pass.
- The `avg_exams` and the `avg_hw` must be greater than 70.
- Either `avg_exams` or `avg_hw` must be greater than 85 (or both of them).

Example Function Calls

`grader(72,88,22)` → prints "PASS"

`grader(66,100,24)` → prints "FAIL"

`grader(100,90,18)` → prints "FAIL"

Grade: *5 Task → Each task is 20 points*

If you don't create the required functions with the proper function name, you will loose 5 points for each function.

Deliverables: Create a folder called "**yourblazerid_HW1**" and save all of your project files into this folder. Submit **yourblazerid_HW1.zip** file and the *independence completion form* to the Canvas (do not forget to sign the form).

Good Luck