Object Oriented Programming & Design
CS 244 - 001
Department of Physics and Computer Science
Medgar Evers College
Exam 1

Direction: Submit your typed work in the Exams directory of your github repository and/or as an attachment on Google classroom under the Exam01 assessment. All submissions should have their appropriate extensions.

Section	Maximum Points	Points Earned
Fundamentals	5	
Problem Solving	10	
Debugging	5	
Total	20	

Fundamentals

1. **For each of the following questions, write ONLY what is requested.**

   a. Write a header file that only consists of a header guard with the id `SPADE_H`

   b. Define a double function named `SumOfProducts()` that takes two double parameters and returns the sum of the squares of the parameters.

   c. For each of the following statements, state if it is true or false

   - volatile (standard) pointers cannot point to constant variables.
   - constant pointers can point to nonconstant variables.
   - readonly pointers need to be initialized when declared.
   - constant pointers cannot point to nonconstant variables.

   d. Given that an int variable named c that has been initialized, write a statement(s) that makes c even only if it is odd.

   e. Write a statement(s) that displays odd multiples of 7 between 1 and 1000 inclusively.

Problem Solving

2. **Morse code is a method to encode telecommunication messages. The morse code for each letter is**

A	●–	**B**	–●●●
C	–●–●	**D**	–●●
E	●	**F**	●●–●
G	––●	**H**	●●●●
I	●●	**J**	●–––
K	–●–	**L**	●–●●
M	––	**N**	–●
O	–––	**P**	●––●
Q	––●–	**R**	●–●
S	●●●	**T**	–
U	●●–	**V**	●●●–
W	●––	**X**	–●●–
Y	–●––	**Z**	––●●

Your objective is translate the letters of a word to morse code encoding. You must write a string function named `Translate()` that a string parameter. It will return a string of the parameter translated into morse code with a space between each code. Furthermore, use asterisks for the dots. The function should be case-insensitive (the case of letters does not matter) and it should do nothing for all the other characters in the string parameter. For instance, the table below show returns for various calls

Call	Return
`Translate("hey")`	`"**** * -*--"`
`Translate("3RD")`	`"*-* -**"`
`Translate("p5Hw91q")`	`"*--* **** *-- --*-"`

3. **A statistician dealing with a dataset of 100 numbers would like to calculate the variance of the dataset, which is**

$$variance = \frac{\sum_{i=1}^{n} (d_i - \mu)^2}{n}$$

where n is the size of the dataset, μ is the average of the dataset, d_is are the values of the dataset. In other words, the variance is the sum of the squares of the difference of each value minus the average all divided by the size of the dataset.

Your objective is to write a double function named `Variance()` that takes a double array parameter. Assuming that the parameter represents a dataset of 100 numbers, the function will return the variance of the dataset.

Debugging

4. **For each function below, write ONLY the line number and the entire corrected line for each line that contains an error or prevents the function from executing the provided intent. Perform the minium amount of changes to the functions.**

a. `/*Intent: it returns 'g' if the parameter is a multiple of 3, 'e' if the parameter is a multiple of 7, 'z' if the parameter is a multiple of both 3 and 7, or '0' otherwise*/`

```
01 | char C(double n)
02 | {
03 |   if(n % 3 == 0)
04 |   {
05 |     return 'z';
06 |   }
07 |   else if(n % 7 == 0)
08 |   {
09 |     return 'g';
10 |   }
11 |   else if(n % 3 == 0 && n % 7 == 0)
12 |   {
13 |     return 'g';
14 |   }
15 |   return '0';
16 | }
```

b. `/*Intent: it adds the value of the first parameter to the second parameter and returns the old value of the second parameter*/`

```
01 | double I(const double& a,const double& b)
02 | {
03 |   int c = b;
04 |   a += b;
05 |   return c;
06 |
```

c. `/*Intent: prompts and reads in a first and last name separately, and then, returns a string of the inputs in the format:"first-last"*/`

```
01 | string F()
02 | {
03 |   String first, last;
04 |
05 |   cout << 'Enter first name:  ';
06 |   cin >> first;
07 |   cout << "Enter last name:  ";
08 |   cin >> last;
09 |   cout << (first + "-" + last);
10 | }
```