Activity No. 1.3	
Hands-on Activity 1.3: Writing First Program using C++ Language	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 9/1**
Section: CPE11S1	**Date Submitted: 9/1**
Name(s): Juan Paulo C. Lara	**Instructor: Engr. Jimlord M. Quejado**
6. Output	
7. Supplementary Activity	

Exercise 4.1 : Try to create a simple program using C++ language that outputs your whole name. Using the new line syntax, output your program and the course and section.

```cpp
#include <iostream>
using namespace std;
int main(){
    cout << "Juan Paulo C. Lara\n";
    cout << "CPE007\n";
    cout << "CPE11S1";
    return 0;
}
```

```
Output
Juan Paulo C. Lara
CPE007
CPE11S1

=== Code Execution Successful ===
```

Exercise 4.2: Write a program in the "C++" language that prints your name 3 times. Remember to include a return statement and make proper use of the main function.

```cpp
#include <iostream>
using namespace std;
int main(){
    cout << "Juan Paulo C. Lara\n";
    cout << "Juan Paulo C. Lara\n";
    cout << "Juan Paulo C. Lara";
    return 0;
}
```

```
Output
Juan Paulo C. Lara
Juan Paulo C. Lara
Juan Paulo C. Lara
```

Activities:

1. Check the program below. Find all possible compilation errors and logic errors. Fix them. Your version of the program must print the same result as the expected output. Before you use your compiler, try to find the errors only by manual code analysis.

Example Output:

The value of five is: 5

```cpp
#include <iostream>
        int main()
        {
        cout("The value of five is:"<< 5int);
        return 0;
        }
```

The first error that comes to mind is the use of the parenthesis after the cout command. The cout command is supposed to have a “<<” inserting the string or variable to output the information inputted at the line. Second is the “5int” before the parenthesis. This is where the clue of the int command assigning the value 5 as an output to the phrase “The value of five is: “ comes from.
Fixed:

```
#include <iostream>
int main(){
    int five = 5;
    std::cout << "The value of five is: " << five << std::endl;
    return 0;
}
```

```
Output
The value of five is: 5


=== Code Execution Successful ===
```

2. Check the program below. Find all possible compilation errors and logic errors. Fix them. Your version of the program must print the same result as the expected output. Before you use your compiler, try to find the errors only by manual code analysis.
Example output
The value of six is: 6

```
int main()
{
cout<<"The value of six is:"<<16,0-10-;
return 0;
}
```

One of the first noticeable errors is the comma after number 16. Next is the minus sign after 10, which has nothing after it. And another is the lack of “std::” or “using namespace std;”. Lastly and most importantly, the lack of the header “#include <iostream>. Not only does the code not state where to refer to the data streaming commands, it also does not correctly tell how to calculate the value of six.
Fixed:

```
#include <iostream>
using namespace std;
int main(){
    cout << "The value of six is: " << 16-10 << endl;
    return 0;
}
```

```
Output
The value of six is: 6


=== Code Execution Successful ===
```

3. Check the program below. Find all possible compilation errors and logic errors. Fix them. Your version of the program must print the same result as the expected output. Before you use your compiler, try to find the errors only by manual code analysis. If you want to improve the variable names, then do so, but remember that variable names have to be as descriptive as possible, and also as short as possible.
Example output
The value of ten is: 10

```
#include <iostream>
using namespace std;

int main()
{
int simpleVariable = 10;
cout<<"The value of ten is:"<<otherVariable);
return 0;
}
```

The first noticeable error is the use of a wrong variable in the cout line, which is “otherVariable” where the declared variable is “simpleVariable”. Since it says simple and other at the following line, it obviously means that a wrong variable is used instead of the stated variable above. Another is the parenthesis right after the already wrong variable and before the terminating statement.
Fixed:

```
#include <iostream>
using namespace std;
int main(){
    int simpleVariable = 10;
    cout<<"The value of ten is: "<< simpleVariable << endl;
    return 0;
}
```

```
Output

The value of ten is: 10


=== Code Execution Successful ===
```

4. Check the program below. Find all possible compilation errors and logic errors. Fix them. Your version of the program must print the same result as the expected output. Before you use your compiler, try to find the errors only by manual code analysis. If you want to improve the variable names, then do so, but remember that variable names have to be as descriptive as possible, and also as short as possible.
Example output
One hour is 3600 seconds

```
#include <iostream>
using namespace std;

int main()
{
int 60seconds = 60;
int 60minutes = 50;
cout<<"One hour is "<<60seconds * 60minutes);
return 0;
}
```

One of the noticeable errors is the invalid variables “60seconds” and “60minutes” that start with numbers. Another is the lack of the phrase “ seconds” after the multiplication operation at the line. Also, there is a typo at the second variable “60minutes” which is stated to be 50 when it should be 60.
Fixed:

```cpp
#include <iostream>
using namespace std;
int main(){
    int sixtySeconds = 60;
    int sixtyMinutes = 60;
    cout << "One hour is " << sixtySeconds * sixtyMinutes << "
        seconds." << endl;
    return 0;
}
```

Output

```
One hour is 3600 seconds.


=== Code Execution Successful ===
```

5. Check the program below. Find all possible compilation errors and logic errors. Fix them. Your version of the program must print the same result as the expected output. Before you use your compiler, try to find the errors only by manual code analysis. If you want to improve the variable names, then do so, but remember that variable names have to be as descriptive as possible, and also as short as possible.
Example output
Localhost IP is 127.0.0.1

```cpp
#include <iostream>
using namespace std;

int main()
{
int ip Part1 = 027;
int ip Part2 = 0;
int ip Part3 = 0;
int ip Part4 = 1;
cout<<"Localhost IP is "<< ip Part1, ip Part2, ip Part3, ip Part4);
}
```

The errors are the commas after each variable to output instead of using the “<<” operator. The spaces of the stated variables are also an invalid syntax since the language will assume it’s “ip” but since it sees “ip” four times, it will throw up a compilation error.
Fixed:

```cpp
#include <iostream>
using namespace std;
int main(){
    int ipPart1 = 127;
    int ipPart2 = 0;
    int ipPart3 = 0;
    int ipPart4 = 1;
    cout << "Localhost IP is " << ipPart1 << "," << ipPart2 << "."
        << ipPart3 << "." << ipPart4 << endl;
    return 0;
}
```

```
Output

Localhost IP is 127,0.0.1


=== Code Execution Successful ===
```

8. Conclusion

Overall, this activity further teaches me how to read possible errors in the codes given and how to correct them without even writing them into the IDE just yet. The previous class discussions also taught me how to find ways to quickly spot errors and know how arithmetic operations happen in code before execution. As preparation for the upcoming preliminary examinations of this course, the fundamentals should always be the first to learn as this is where it leads me to extend my understanding of the C++ language.

9. Assessment Rubric