

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: 09/01/25
Section: CPE11S1	Date Submitted: 09/02/25
Name(s): Angel Mae Ramirez	Instructor: Engr. Jimlord Quejado

6. Output

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.

CODE:

```
#include <iostream>
int main() {
    std::cout << "Angel Mae Ramirez" << std::endl;
    std::cout << "CPE007" << std::endl;
    std::cout << "CPE11S1" << std::endl;
    return 0;
}
```

```
main.cpp [ ] ⓘ Share Run
1 #include <iostream>
2 int main() {
3     std::cout << "Angel Mae Ramirez" << std::endl;
4     std::cout << "CPE007" << std::endl;
5     std::cout << "CPE11S1" << std::endl;
6     return 0;
7 }
```

RESULT:

Output	Clear
Angel Mae Ramirez 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.

CODE:

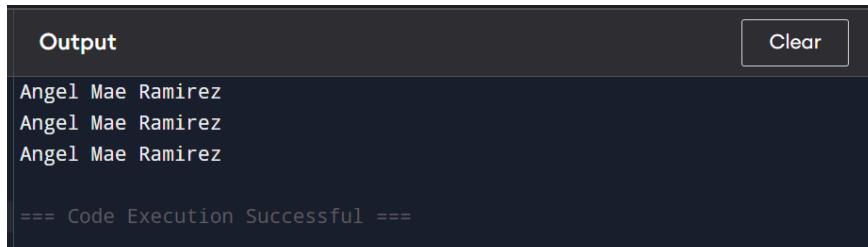
```
#include <iostream>
int main() {
    std::cout << "Angel Mae Ramirez" << std::endl;
    std::cout << "Angel Mae Ramirez" << std::endl;
    std::cout << "Angel Mae Ramirez";
    return 0;
}
```

}



```
main.cpp
1 #include <iostream>
2 int main() {
3     std::cout << "Angel Mae Ramirez" << std::endl;
4     std::cout << "Angel Mae Ramirez" << std::endl;
5     std::cout << "Angel Mae Ramirez";
6     return 0;
7 }
```

RESULT:



Output

Angel Mae Ramirez
Angel Mae Ramirez
Angel Mae Ramirez

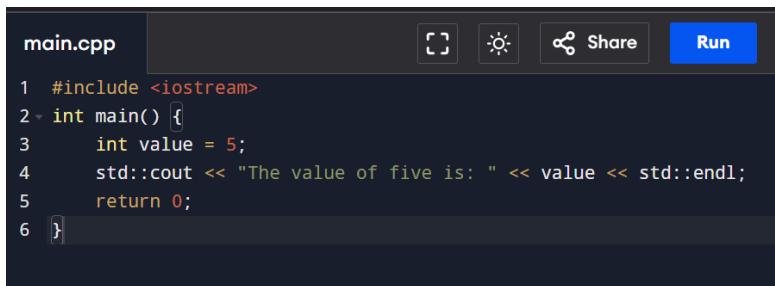
==== Code Execution Successful ===

7. Supplementary Activity

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.

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

FIXED VERSION:



```
main.cpp
1 #include <iostream>
2 int main() {
3     int value = 5;
4     std::cout << "The value of five is: " << value << std::endl;
5     return 0;
6 }
```

RESULT:

The screenshot shows a dark-themed code editor or terminal window. At the top left is a tab labeled "Output". At the top right is a "Clear" button. The main area contains the following text:
The value of five is: 5
== Code Execution Successful ==

ERRORS:

The common errors that I found in the provided code include missing semicolons and the incorrect use of cout without the std :: prefix. Additionally, the code might try to print the value before it's assigned, which is a logic error. Variables must be assigned a value before being used. For example, if the variable vslue is declared but initialized after the cout statement, it results in unexpected behavior.

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.

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

FIXED VERSION:

The screenshot shows a code editor with a file named "main.cpp". The code is as follows:
1 #include <iostream>
2 int main() {
3 int six = 6;
4 std::cout << "The value of six is: " << six << std::endl;
5 return 0;
6 }

RESULT:

The screenshot shows a dark-themed code editor or terminal window. At the top left is a tab labeled "Output". At the top right is a "Clear" button. The main area contains the following text:
The value of six is: 6
== Code Execution Successful ==

ERRORS:

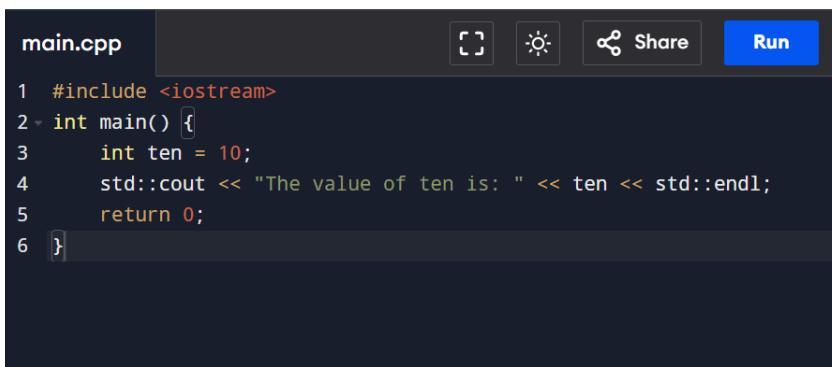
One likely error that I see in the original code is assigning a string "6" to an integer variable, which causes a type mismatch. In C++, integer variables should be assigned numeric values without quotation marks. Another common issue is missing the semicolon at the end of the return statement or the output line.

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.

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

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

FIXED VERSION:

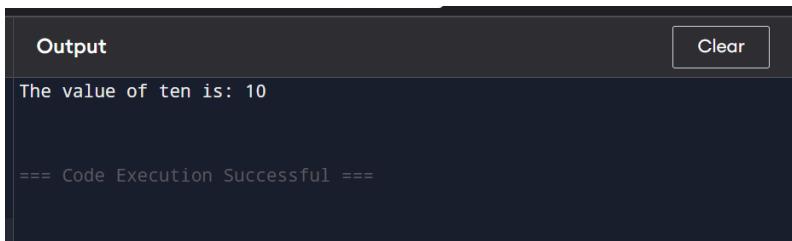


The screenshot shows a code editor window titled "main.cpp". The code is as follows:

```
1 #include <iostream>
2 int main() {
3     int ten = 10;
4     std::cout << "The value of ten is: " << ten << std::endl;
5     return 0;
6 }
```

The code is displayed in a monospaced font, with syntax highlighting for keywords like "int" and "std::cout". The editor has a dark theme with light-colored text. At the top, there are icons for file operations, a share button, and a "Run" button.

RESULT:



The screenshot shows the execution results in a terminal-like interface. The output is:

```
Output
Clear
The value of ten is: 10
==== Code Execution Successful ===
```

The output is displayed in a monospaced font, with the "Output" label and "Clear" button visible at the top.

ERRORS:

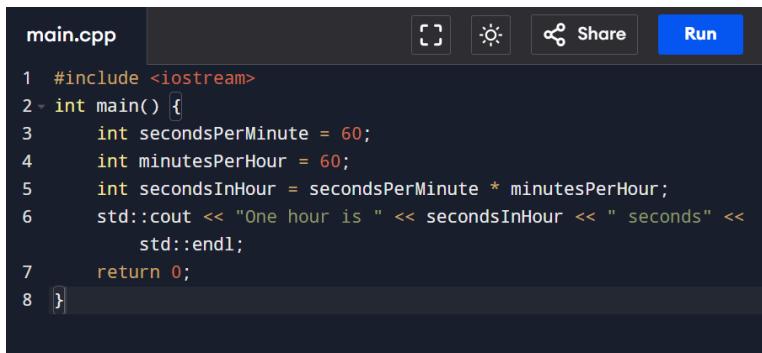
A typical mistake here is trying to name a variable 10, which is not allowed because variable names cannot start with a digit or be a number. This would cause a syntax error. Another possible issue is using the literal number 10 directly in the output without explanation.

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.

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

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

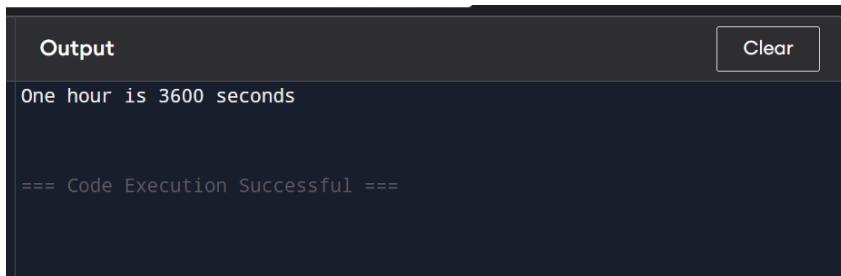
FIXED VERSION:



The screenshot shows a code editor window titled "main.cpp". The code has been modified to use descriptive variable names and correct syntax. It calculates the total seconds in an hour by multiplying the number of seconds per minute (60) by the number of minutes per hour (60), and then outputs the result as "One hour is 3600 seconds".

```
1 #include <iostream>
2 int main() {
3     int secondsPerMinute = 60;
4     int minutesPerHour = 60;
5     int secondsInHour = secondsPerMinute * minutesPerHour;
6     std::cout << "One hour is " << secondsInHour << " seconds" <<
7         std::endl;
8 }
```

RESULT:



The screenshot shows a terminal window with the title "Output". The output of the program is displayed, showing the calculation "One hour is 3600 seconds" followed by a success message "==== Code Execution Successful ===".

```
Output
Clear
One hour is 3600 seconds
==== Code Execution Successful ===
```

ERRORS:

The common errors include missing semicolons at the end of lines and using unclear variable names. For example, the variables might be named simply minutes and seconds, which could confuse their meaning. It's better to use descriptive names like secondsPerMinute and minutesPerHour.

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.

```

#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);
}

```

FIXED VERSION:

The screenshot shows a code editor window titled "main.cpp". The code is as follows:

```

1 #include <iostream>
2 int main() {
3     std::cout << "Localhost IP is 127.0.0.1" << std::endl;
4     return 0;
5 }

```

At the top of the window are several icons: a file icon, a copy icon, a share icon, and a "Run" button.

RESULT:

The screenshot shows a terminal window with the tab "Output" selected. The output text is:

```

Output
Clear
localhost IP is 127.0.0.1
==== Code Execution Successful ====

```

ERRORS:

A frequent issue is writing this output text without enclosing it in quotation marks. In C++, strings that are printed with cout must be enclosed in double quotes, otherwise the compiler will think you're referring to variables or unknown identifiers, which leads to errors. Another mistake is omitting the semicolon at the end of the output statement.

8. Conclusion

In this activity, I learned how to write simple programs in C++ and how to fix errors in codes. At first, it was kinda confusing specially with all the semicolon stuff and how you need to always put std:: before cout, but after doing the exercises I understand it better now. I also learned that you have to always assign values before using them and to not forget the quotation marks when printing words or sentences. Debugging the codes help me to notice small mistakes that can stop the program from working. Sometimes I forget to put semicolon or I write the variable name wrong, and now I know that even small mistake can break the program. This activity also helped me to practice how to read codes and understand what it's trying to do before fixing it. Overall, I feel more confident now with using C++ and I hope to get better with more practice.

9. Assessment Rubric

Rubric for SO 7 (6)

Criteria	Ratings						Pts
© SO 7 PI 1 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently	4 pts Satisfactory Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor Relies on classroom instruction only	1 pts Very Poor No initiative or interest in acquiring new knowledge	6 pts
© SO 7 PI 2 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Completes an assigned task independently and practices continuous improvement	5 pts Good Completes an assigned task without supervision or guidance	4 pts Satisfactory Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory Requires detailed or step-by-step instructions to complete a task	2 pts Poor Shows little interest to complete a task independently	1 pts Very Poor No interest to complete a task independently	6 pts
© SO 7 PI 3 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory Apply the gathered information to formulate the problem	2 pts Poor Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor Gather information from a variety of sources	6 pts
© SO 7 PI 4 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory Shows some creative ways to solve the problem	2 pts Poor Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor Ideas are copied or restated from the sources consulted	6 pts

Total Points: 24