

## Hands-on Activity 2.1

### Data Types and Arithmetic Operations

<b>Course Code:</b> CPE007	<b>Program:</b> Computer Engineering
<b>Course Title:</b> Programming Logic and Design	<b>Date Performed:</b> 8/4/25
<b>Section:</b> CPE11S1	<b>Date Submitted:</b> 8/4/25
<b>Name(s):</b> Paula Esguerra	<b>Instructor:</b> Engr. Jimlord Quejado

#### 6. Output

**Example 1:** The following program has an output of:

The value of seven is: 7.000000

The value of eight and a half is: 8.500000

Can you find all possible compilation errors and logic errors? Can you fix them to 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>
```

```
using namespace std;
int main()
{
    cout<<"The value of seven is: ";
    cout<<"The value of eight and a half is: ", <<8.5;
return 0;
}
```

The screenshot shows a code editor window with the following details:

- Title Bar:** The file name "main.cpp" is displayed in the top-left corner.
- Toolbars:** A toolbar at the top right contains icons for copy, paste, share, and run.
- Code Area:** The main content area displays the C++ code. Line numbers 1 through 13 are shown on the left. The code includes an include directive, a main function, and two cout statements. The second cout statement has a syntax error where the decimal point is missing from the floating-point literal "8.5".

**Output****Clear**

```
The value of seven is: The value of eight and a half is: 8.5
```

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

The logic error was (,) then adding << std::endl; to have enter and space the output

**Example 2:** The following program has an output of:

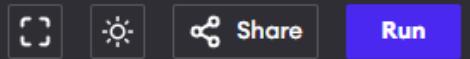
```
The value of seven is: 7.000000
```

```
The value of eight and a half is: 8.500000
```

Can you find all possible compilation errors and logic errors? Can you fix them to 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>
using namespace std;
int main()
{
    cout<<"The value of seven is: "<< 7.0;
    cout<<"The value of eight and a half is: "<<8.5;
return 0;
}
```

main.cpp



```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout<<"The value of seven is: "<< 7.0 << std::endl;
6     std::cout<<"The value of eight and a half is: "<<8.5;
7 return 0;
8 }
9
```

## Output

Clear

```
The value of seven is: 7
The value of eight and a half is: 8.5
```

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

The logic error was the number was lacking a decimal point to continue, then adding `<< std::endl;` to enter / space the output.

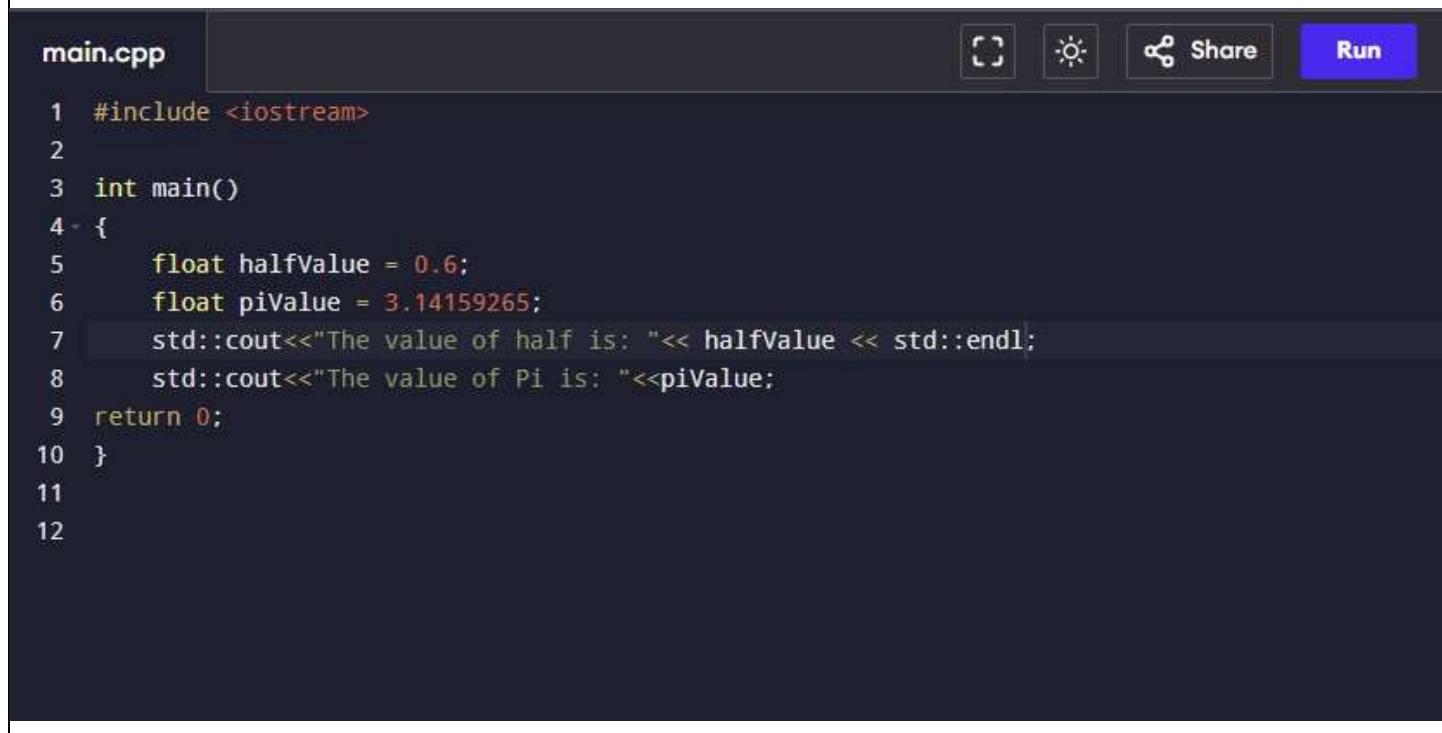
**Example 3:** The following program has an output of:

The value of half is: 0.500000

The value of Pi is: 3.141593

Can you find all possible compilation errors and logic errors? Can you fix them to 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()
{
    float halfValue = 0.6;
    float piValue = 3.141 592 65;
    cout<<"The value of half is: "<< half Value;
    cout<<"The value of Pi is: "<<pi_Value;
return 0;
}
```



The screenshot shows a code editor interface with a dark theme. The file tab at the top left is labeled "main.cpp". To its right is a toolbar with icons for copy, paste, run, and share. The main area contains the C++ code for Example 3. The code uses floating-point literals with decimal points instead of commas. The line "cout<<"half Value;" is highlighted in yellow, indicating a spelling error. The code is numbered from 1 to 12 on the left. The background of the code editor is dark, and the text is white or light gray.

```
1 #include <iostream>
2
3 int main()
4 {
5     float halfValue = 0.6;
6     float piValue = 3.14159265;
7     std::cout<<"The value of half is: "<< halfValue << std::endl;
8     std::cout<<"The value of Pi is: "<<piValue;
9 return 0;
10 }
11
12
```

### Output

Clear

The value of half is: 0.6

The value of Pi is: 3.14159

==== Code Execution Successful ===

The logic error was the number of pi has space that made the error visible, error in case sensitive titles then adding << std::endl.

#### Example 4: Sample program for Adding Two Integers

```
#include <iostream>
int main()
{
    int integer1, integer2, sum; /*declaration */
    cout<<"Enter first integer: \n"; /* prompt */
    cin>>integer1; /* read an integer */
    cout<<"Enter second integer: \n"; /* prompt */
    cin<<integer2; /* read an integer */
    sum = integer1 + integer2; /* assignment of sum */
    cout<<"Sum is : "<<sum; /* print sum */

return 0; /* indicate that program ended successfully */
}
```

A screenshot of a code editor window titled "main.cpp". The code editor displays the C++ program from above. The code is color-coded: comments are in gray, strings are in red, and variables and functions are in black. The editor has a dark theme. At the top right, there are several icons: a copy icon, a settings icon, a share icon, and a blue "Run" button. Below the code editor, there is a large white area which likely represents the terminal or output window where the program's output would be displayed.

```
1 #include <iostream>
2 int main()
3 {
4     int integer1, integer2, sum; /*declaration */
5     std::cout<< "Enter first integer: \n" ; /* prompt */
6     std::cin>>integer1; /* read an integer */
7     std::cout<< "Enter second integer: \n" ; /* prompt */
8     std::cin>>integer2; /* read an integer */
9     sum = integer1 + integer2; /* assignment of sum */
10    std::cout<< "Sum is : "<<sum; /* print sum */
11
12 return 0; /* indicate that program ended successfully */
13 }
14
15
```

**Output****Clear**

```
Enter first integer:  
5  
Enter second integer:  
8  
Sum is : 13  
  
==== Code Execution Successful ===
```

The logic error was the lack of std::, and change 8's "<< "to ">>" and wrong quotation.

**7. Supplementary Activity**

1. Take a look at the code below: it assigns two integer values, manipulates them and finally outputs the result and bigresult variables. The problem is that the manipulations have been described using natural language, so the code is completely useless now. Act as an intelligent (naturally!) compiler and translate the formula into a real "C" code notation. Test your code using the data provided.

Programs C++ Online Compiler

<pre> 1. #include&lt;iostream&gt; 2. 3 int main(void) 4 { 5 6     int xValue= 5; 7     int yValue=9; 8     int result; 9     int bigResult; 10 11    xValue += 3; 12    yValue -= xValue; 13    result = xValue * yValue; 14    result += result; 15    result -= 1; 16    yValue = result % result; 17    result += result + xValue; 18    bigResult = result * result * result; 19    result += xValue * yValue; 20 21    std::cout&lt;&lt;"result: "&lt;&lt;result &lt;&lt; std::endl; 22 23    std::cout&lt;&lt;"big result: "&lt;&lt; bigResult; 24 25    return 0; 26 27 28 } </pre>	  	<b>Run</b>	<b>Output</b> <pre> result: 38 big result: 54872 --- Code Execution Successful --- </pre>
--	---	------------	--

2. Complete the program below. Compute the accrued amount of money with a starting value of 100 and an annual interest rate of 1.5%. Compute and print the results for first three years. Your version of the program must print the same result as the expected output for every year. Compute each annual value on the basis of the previous year's value.

<pre> 1 // Online C++ compiler to run C++ program online 2 #include &lt;iostream&gt; 3 #include &lt;iomanip&gt; // Required for std::setprecision 4 int main() { 5 6     float startValue = 100; 7 8     float interestRate = 0.015; 9 10    float firstYearValue; 11 12    float secondYearValue; 13 14    float thirdYearValue; 15 16    firstYearValue = startValue * (1 + interestRate); 17    secondYearValue = firstYearValue * (1 + interestRate); 18    thirdYearValue = secondYearValue * (1 + interestRate); 19 20    // Set output fixed-point notation and 6 decimal places 21    std::cout &lt;&lt; std::fixed &lt;&lt; std::setprecision(6); 22 23    std::cout &lt;&lt; "After first year:" &lt;&lt;firstYearValue &lt;&lt; std::endl; 24    std::cout &lt;&lt; "After second year:" &lt;&lt;secondYearValue &lt;&lt; std::endl; 25    std::cout &lt;&lt; "After third year: " &lt;&lt;thirdYearValue &lt;&lt; std::endl; 26 27    return 0; 28 } </pre>	  	<b>Run</b>	<b>Output</b> <pre> After first year:101.500000 After second year:103.022499 After third year: 104.567833 --- Code Execution Successful --- </pre>
---	---	------------	---

## 8. Conclusion

Finding logic errors are a bit of hard to understand but when you think carefully about the symbols, you can change a bit in your coding, then the coding will be fixed. First you need to understand the error, if you don't understand then search in Google, the arrows show the error is, sometimes it can be the wrong signs or symbols. And then, though using

"using namespace std" is much easier, but I do most prefer not using "using namespace std". it is important to always double check the coding before running through. I did learned something new, to acquire 6 decimal places, you have to add, <iomanip> and it's relations.

## 9. Assessment Rubric

Rubric for SO 7 (6)							
Criteria				Ratings			Pts
④ 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 pt
④ 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 pt
④ 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 pt
④ 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 pt

Total Points: 2