

Activity #1: Introduction to C++

Recorder's Report

Manager:


Reader:

Recorder:

Driver:

Date:

Score: Satisfactory / Not Satisfactory

Record your team's answers to the key questions (marked with ) below.

(a) Model 1, Question #5

(b) Model 2, Question #9

(c) Model 3, Question #14

Activity #1: Introduction to C++

In this course, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to the process of analyzing an algorithm complexity.

Content Learning Objectives

After completing this activity, students should be able to:

- Explain how to print content to the screen using C++
- Explain how to create a comment in C++ code
- Determine the difference between a *string literal* and a *integer*
- Explain how to input data into a variable in C++
- Explain the meaning and purpose of a variable

Process Skill Goals

During the activity, students should make progress toward:

- Create input and output statements in C++
- Create C++ code that displays the results to calculated addition facts
- Create C++ code that prompts the user for data and stores it in a variable
- Select valid and meaningful variable names
- Discuss problems and programs with all group members



Preston Carman derived this work from Lisa Olivieri work found at <https://www.dropbox.com/sh/2fx6pg4ydp9t7x/AAAdJfzvLjeym1gJwKrIWwhBa?preview=Python+Activity+01+Introduction+to+Python++POGIL.docx> and continues to be licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Model 1 A C++ Program

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

int main() {
    cout << "Go!\n";
}
```

Refer to Model 1 above as your group develops consensus answers to the questions below.

Questions (20 min)

Start time:

1. The file activity01a.cpp contains this C++ program. Run it and determine what it does.
2. Replace line 5 with each of the following. What is produced? Indicate if there is a problem.
 - a) `cout << "Hello, my name is Jon";`
 - b) `cout << Hello, my name is Jon;`
 - c) `cout << "Hello,\nmy name is Jon";`
3. A *string literal* is a sequence of characters surrounded by double quotation marks (" "). Use this term to describe the difference between part (a) and part (b) above.

4. What caused the different output format for statements (a) and (c) in question 2?

5. What do you think the following C++ statements output?



a) `cout << 2+5;`

b) `cout << 2*5;`

c) `cout << "2+5";`

d) `cout << "Age:" << 20;`

6. With reference to the output for each statement in question 5,

a) What is the difference in the output for statements (a) and (c)?

b) What caused the difference?

c) Which statement(s) contained a *string literal*? Explain.

d) What does the extra `<<` do in part (d) of question 4? How does it affect the spacing of the output?

Model 2 Another C++ Program

```
// Programmer: Jane Doe
// Date: 9/19/2019
// Description: Prints a welcome statement
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, Jane\n";
    cout << "Enjoy programming in C++\n";
}
```

Output:

Hello, Jane
Enjoy programming in C++

Refer to Model 2 above as your group develops consensus answers to the questions below.

Questions (10 min)

Start time:

7. What do the first three lines of this program do?
8. What would happen if placed `//` in front of the code: `cout << "Hello, Janen"` in this program?
9. A *comment* is text in a program that explains or annotates the source code, but does not affect how the program runs. Complete the statements on lines 6-7 so that the C++ program below produces the desired output. Add a comment on line 4. This code is in the file `activity01b.cpp`



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

int main() {

    cout <<
    cout <<
    cout <<

}
```

Output:

Congratulations!
You just created
your first C++ program

Model 3 Collecting Input

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

int main() {
    string name;
    cout << "What is your name? ";
    cin >> name;
    cout << "Your name is " << name;
}
```

Refer to Model 3 above as your group develops consensus answers to the questions below.

Questions (20 min)

Start time:

10. You will find this code in the file `activity01c.cpp`. Execute it and determine what the program does.

11. The word *name* in this code identifies a *variable* (a name given to a memory location used to store data). What happens to the data that the user of this program enters?

12. In C++, each variable has a *type* which specifies what sort of data it can store. What is the type of the variable in the program above? What sort of data can it store?

13. Suppose you wished to also store the users age. Explain the errors that occur when you attempt to create the following *integer* (int) variables to this program.

a) `int age?;`

- b) `int your age;`
- c) `int 2age;`
- d) `int int;`
- e) `int the.age;`

14. The following are valid variable names for age. Based on this list, and the errors you found above, write two rules that valid variable names must follow in C++.



`age age2 myAge the_age`

15. Suppose you need a variable to store the cost of an item. The following names are suggested. Are they valid? Are they good choices?

- a) `price`
- b) `priceoftheitem`
- c) `x`
- d) `itemPrice`

16. Modify the C++ program found in `activity01c.cpp` so that it prompts the user for two integers and then prints out the sum of those two integers as shown in the example output below. Use meaningful variable names and comments in your code.

Output:

```
Enter first number: 7
Enter second number: 12
7 + 12 = 19
```