Class Activity 1.3: User input and Expressions

In this activity, we will review retrieving user input and expressions in C++.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: _____ Date: _____

Role	Team Member Name
Manager. Makes sure team starts quickly and remain focused during the activity; takes care of time management; makes sure all voices in the team are heard	
Presenter. The only person in the team assigned to communicate questions and clarifications with the instructor or other teams; ensures all team members have had a chance to respond before asking outside sources; ensures that everyone in the team agrees on what to ask if an outside source is needed; presents conclusions of the team to the class, as requested.	
Reflector. Guides consensus-building process so that the team agrees on responses to questions; observes team dynamics and behavior with respect to the learning process; reports to the team periodically during the activity on how the team performs; possibly report to the entire class about how well the team is operating.	
Recorder. Records the names and roles of the group members at the beginning of each activity; records the important aspects of group discussions, observations, insights, etc.; the recorder's report is a log of the important concepts that the group has learned.	

1. Create a program that asks the user to input their first name, middle name, and last name. The program should then use the user's input and display it in this format: `lastname, firstname middle name` (5 pts)

Output:

```
Please enter your first name: Tuffy
Please enter your middle name: The
Please enter your last name: Titan

Formatted name: Titan, Tuffy The
```

Code:

```
#include <iostream>
int main() {
  std::string first_name = "";
  std::string middle_name = "";
  std::string last_name = "";
  std::cout << "Please enter your first name: ";
  std::cin >> first_name;

  std::cout << "Please enter your middle name: ";
  std::cin >> middle_name;

  std::cout << "Please enter your last name: ";
  std::cin >> last_name;

  std::cout << "Formatted name: " << last_name << ", " <<
first_name
            << " " << middle_name << "\n";
  return 0;
}
```

2. Trace the following program and predict the screen output. (2 pts)

Code:

```
#include <iostream>
#include <iomanip>
int main() {
  int x1 = 3;
  int x2 = 7;
  int y1 = 0;
  int y2 = 12;
  std::cout << "The slope of the line is: " << y2 - y1 / x2 - x1;
  return 0;
}
```

Output:

```
The slope of the line is: 9
```

3. Did the program correctly apply the formula for the slope of a line: $m = \frac{y_2 - y_1}{x_2 - x_1}$? Type in yes in the text box below if it does, otherwise provide a correct version of the *std::cout* statement.

Answer:

```
std::cout << "The slope of the line is: " << (y2 - y1) / (x2 - x1);
```

4. What will *x* contain after the assignment operation?

Code snippet:

```
int x = 25.3;
```

Value: 25

5. What will *y* contain after the assignment operation?

Code snippet:
```
double y = 5;
```

Value: 5.0

6. Create a complete program that will compute the area of a trapezoid using the following formula: $area = \frac{b_1+b_2}{2}h$. The program should ask the user to provide the values for the two bases and the height of the trapezoid then display it's area. Please use meaningful variable names instead of *b1*, *b2*, and *h* to make your code more readable.

Sample Output:
```
Please enter base1 of the trapezoid: 3
Please enter base2 of the trapezoid: 6
Please enter the height of the trapezoid: 5
The area of the trapezoid is: 22.5
```

Code:
```
#include <iostream>
int main() {
  double base1 = 0;
  double base2 = 0;
  double height = 0;
  std::cout << "Please enter base1 of the trapezoid: ";
  std::cin >> base1;
  std::cout << "Please enter base2 of the trapezoid: ";
  std::cin >> base2;
  std::cout << "Please enter the height of the trapezoid: ";
  std::cin >> height;
  double area = (base1 + base2) / 2 * height;
  std::cout << "The area of the trapezoid is: " << area << "\n";
  return 0;
}
```

7. What are constants? Give an example and explain why they are important.

Answer:

Constants are variables whose value cannot be changed during program execution.

const double TAX = 0.075;

They are important because it makes code more readable. It is easier to understand a descriptive variable name than a value that you might forget.

You can also use constants instead of hardcoding values in the code every time. If you change the constant's value, the value changes every part of the code that uses the constant. This is faster than having to search and replace each instance of a value in the code.

8. What will *z* contain after the combined assignment operation?

Code snippet:

```
int z = 3;
z *= 3;
```

Value: 9