Lecture Summary

Anatomy of your first C++ program

Recall the following simple program from first week lectures that demonstrates the basic structure of a C++ program, including libraries, defining the main function, and using the `cout` object for output.

```
#include <iostream>
int main()
{
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

Now, let's break down the different parts of this program to make it understandable:

- `#include <iostream>`: This line is a preprocessor directive that tells the compiler to include the `iostream` library. The `iostream` library provides functionality for input and output operations, like printing text to the screen.
- `int main(){ ... }`: This is the `main` function of your program. Every C++ program must have a `main` function. The program's execution starts from here.
- `std::cout << "Hello, World!"<< std::endl;`: This line uses the cout object from the iostream library to print text to the console. `std::cout` is used to output data to the console, and « is the insertion operator that "inserts" the text or value on its right into the output stream. "Hello, World!" is the text you want to print. `std::endl` is used to insert a newline character and flush the output buffer.
- `return 0;`: This line indicates the end of the `main` function and the successful termination of the program. The value 0 is returned to the operating system, signifying that the program ran without errors.

When you compile and run this program, you'll see the output: `Hello, World!`

Declaring a Variable

In C++, a variable is a named location in memory that stores a value. Before using a variable, you need to declare it, which involves specifying the variable's data type and a name. Here's the general syntax:

```
data_type variable_name;
```

For example, to declare an integer variable named `age`:

```
int age;
```

Assigning a Value to a Variable

After declaring a variable, you can assign a value to it using the assignment operator `=`. The value you assign should match the variable's data type. Here's an example:

```
age = 25;
```

Input/Output using cin and cout

`cin` and `cout` are part of the C++ Standard Library and are used for input and output operations. cin is used for input (reading values from the user), and cout is used for output (displaying values to the user).
Input using cin: To get input from the user, you use cin along with the » operator. You need to specify the variable where the input should be stored. Here's an example of how to read an integer from the user and store it in the age variable:

```
cin >> age;
```

Output using cout: To display output to the user, you use cout along with the « operator. You can output variables, constants, and text. Here's an example:

```
cout << "Your age is: " << age << endl;
```

Lab Questions

1. Create a program `area.cpp` that asks the user for the length and width of a rectangle (both as separate inputs). Calculate and display the area of the rectangle using cout.

2. Develop a program `temperature.cpp` that takes the temperature in Fahrenheit as input and converts it to Celsius using the formula: `Celsius = (Fahrenheit - 32) * 5 / 9`. Output the converted temperature.

3. Write a program `hello.cpp` that requests the user's name as input and then asks for their age. Display a message that includes their name and age, like: `Hello [Name], you are [Age] years old.`

   Hint: `Use std::string` type variable to take input for name. You need to include `string` library by adding the following line before `main` function:

   ```
   #include<string>
   ```