Lecture Summary

Hello World!

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

```cpp
#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 (You can also use `'\n'` character inside the string for new line, e.g., `std::cout << "Hello, World!\n"`).
- **`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.

Setting up `g++` Compiler

Before running this program, you need to compile it using a C++ compiler like `g++`. Verify that you have a C++ compiler installed by running the command: `g++ -version`.
If `g++` is not recognized as a command, you need to install `g++` compiler and/or add the path to the compiler to the PATH environment variable.

- *On your laptop or home computer (Windows)*:

  Follow the guide at `https://www.msys2.org/` to install `MSYS2` and set up the `g++` compiler on your system.

- *In labs*:

  The C++ compiler is already installed on the lab computers. You may need to set the PATH environment variable to run the compiler from the command line.

  Open Windows Command Prompt by pressing `Win + R` and typing `cmd`. Then run the following command in the terminal:

  ```
  set PATH=C:\msys64\ucrt64\bin;%PATH%
  ```

Compiling and Running a C++ Program

Save your program in a text file with `.cpp` extension. To compile and run this program, you can use the following commands:

```
g++ hello.cpp -o hello
./hello
```

where `hello.cpp` is the name of your program file, and `hello` is the name of the executable file generated by the compiler.
If there is an error in your program, the compiler will display an error message indicating the line number and the type of error. For example, if you forget put semicolon after the `cout` statement, you'll get an error like: `error:  expected ';' before 'return'`.
When you compile and run this program, you'll see the output: `Hello, World!`

Declaration and Assignment statements

In C++, we use *variables* to store data. A variable is a named location in memory that stores a value.
Unlike Python or JavaScript, C++ is a *statically typed* language, which means that we need to specify the data type of a variable when we declare it. Some of the common data types in C++ are `int` (integer), `double` (floating-point number), and `std::string` (string). Here's how we declare variables of different types:

```
int age;
double price;
std::string name;
```

To *assign* a value to a variable, you use the assignment operator `=`. For example, you can use variables to store a person's age, a product's price, or a student's grade.

```
age = 25;
price = 19.99;
name = "Alice";
```

A declaration and assignment statement can be combined into a single line:

```
int age = 25;
double price = 19.99;
std::string name = "Alice";
```

Input/Output using `cin` and `cout`

`std::cin` and `std::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:

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

Escape Sequences:

In C++, you can use *escape sequences* to represent special characters in strings.

Escape Sequence	Description
`\n`	newline
`\t`	tab
`\r`	carriage return
`\b`	backspace
`\\`	backslash
`\"`	double quote

For example, the following code will output the text on two lines:

```
std::cout << "Hello, World!\n";
std::cout << "Welcome to C++ Programming\n";
```

Lab Questions

1. Create a program `birthyear.cpp` that asks the user for their name and their year of birth. The program should calculate their age and display a message such as:
   `David, you will be 20 years old this year.`

2. Develop a program `triangle_area.cpp` that takes the length of one side of an equilateral triangle as input and calculates its area. Use the formula:

$$\text{Area} = \left(\frac{\sqrt{3}}{4}\right) \times \text{side} \times \text{side}$$

   Hint: Use a `double` type variable for both the side length and the area. Use the constant value 1.732 instead of $\sqrt{3}$.

3. Write a program `recipe.cpp` that helps calculate the ingredients needed for making a sweet dish. One serving requires 2 cups of sugar and 1 cup of oil. Ask the user how many servings they want to make, and output the total amount of sugar and oil required.

4. Create a program `shopping.cpp` that asks the user to enter the prices of three items they have purchased. The program should calculate the total price including tax. Assume the sales tax is **10%** of the total cost of the three items. Finally, display the total price to the user.

5. Write a program `receipt.cpp` that prints a receipt for the three purchased items from Question 4. The receipt should display the three items, their prices, the tax, and the total cost. The entire receipt should be printed in a single output statement.

   The output should look like this:

```
Item    Price
Item1   10
Item2   20
Item3   30
Tax     6
Total   66
```

6. Write a program `currency.cpp` that asks the user for an amount of money in rupees. The program should calculate how many **100-rupee notes, 50-rupee notes, and 10-rupee notes** are needed to make that amount, using the largest notes first.

   Hints:

   - Integer division (`/`) gives you how many times one number can fit completely into another.
   - The modulo operator (`%`) gives you the remainder left after division.