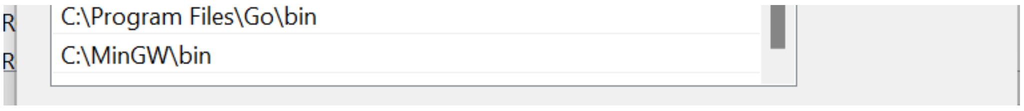
C++ Notes

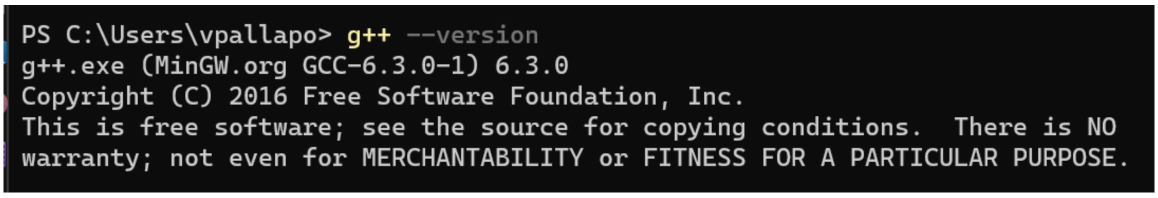
Compiler:

How to install MinGW Compiler?

https://www.youtube.com/watch?v=JsO58opI3SQ



```
C:\Program Files\Go\bin
C:\MinGW\bin
```



```
PS C:\Users\vpallapo> g++ --version
g++.exe (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

IDE Link:

- https://codelite.org/
- VS Code

A workspace is an area that holds projects.

Using the command line interface on windows

To compile the file:

g++ -Wall -std=c++14 main.cpp -o main.exe

-Wall: Enable all compiler warnings

-std: Use C++14 features

g: Produce debugging information

-o: output file name

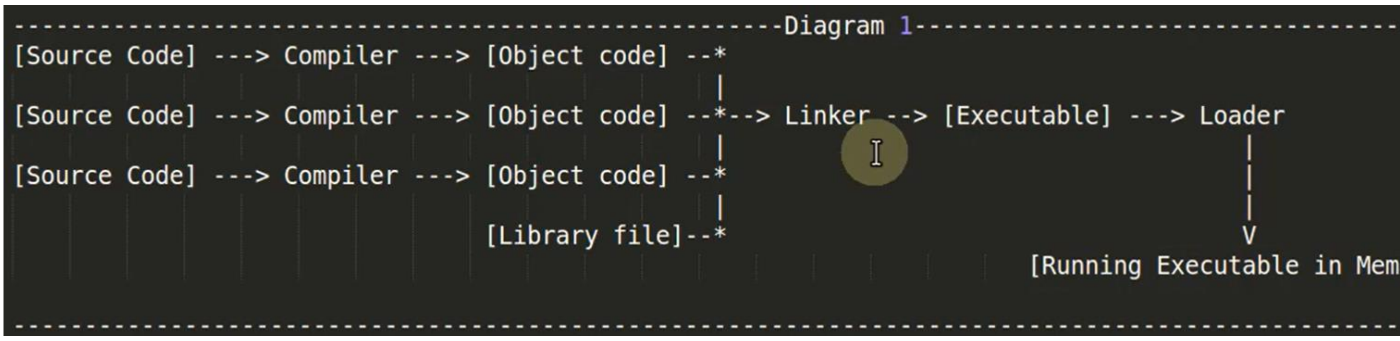
Section 4: Getting Started

Building our first program

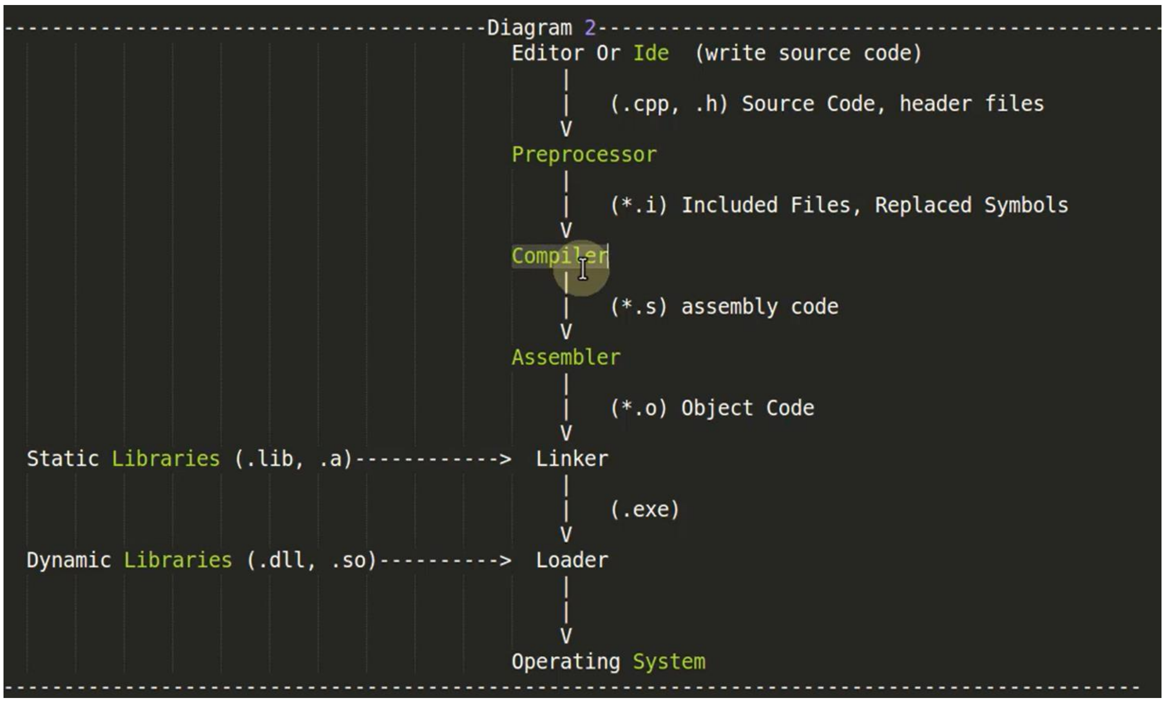
Build = Compile + Link

Clean = cleans all the obj files

Rebuild = Clean + Build



```
------------------------------------------------------Diagram 1----------------------------------
[Source Code] ---> Compiler ---> [Object code] --*
                                                 |
[Source Code] ---> Compiler ---> [Object code] --*--> Linker --> [Executable] ---> Loader
                                                 |                                    |
[Source Code] ---> Compiler ---> [Object code] --*                                    |
                                                 |                                    |
                                 [Library file]--*                                    V
                                                                       [Running Executable in Mem

-------------------------------------------------------------------------------------------------
```



```
--------------------------------------Diagram 2-------------------------------------------
                                       Editor Or Ide  (write source code)
                                           |
                                           |   (.cpp, .h) Source Code, header files
                                           V
                                       Preprocessor
                                           |
                                           |   (*.i) Included Files, Replaced Symbols
                                           V
                                       Compiler
                                           |
                                           |   (*.s) assembly code
                                           V
                                       Assembler
                                           |
                                           |   (*.o) Object Code
                                           V
 Static Libraries (.lib, .a)------------>  Linker
                                           |
                                           |   (.exe)
                                           V
 Dynamic Libraries (.dll, .so)---------->  Loader
                                           |
                                           |
                                           V
                                       Operating System
------------------------------------------------------------------------------------------
```

What are comiler errors?

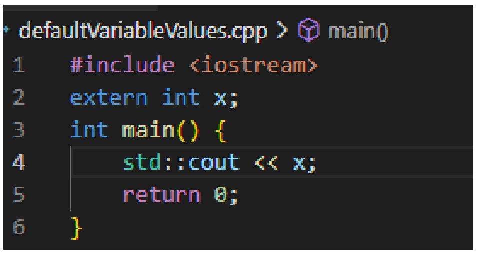
Compiler Warnings

The compiler has recognized an issue with your code that could lead to a potential problem.

Linker Errors

The linker is having trouble linking all the object files together to create an executable.

Usually there is a library or object file that is missing.



defaultVariableValues.cpp > main()

```cpp
#include <iostream>
extern int x;
int main() {
    std::cout << x;
    return 0;
}
```

g++ -Wall -std=c++14 .\defaultVariableValues.cpp -o .\defaultVariableValues.exe

C:/mingw64/bin/../lib/gcc/x86_64-w64-mingw32/12.2.0/../../../../x86_64-w64-mingw32/bin/ld.exe: C:\Users\vpallapo\AppData\Local\Temp\ccvWunD1.o:defaultVariableValues.cpp:(.rdata$.refptr.x[.refptr.x]+0x0): undefined reference to `x'

Runtime Errors

Erros that occur when the program is executing.

Some typical runtime errors

- Dived by zero
- File not found
- Out of memory

Can cause you program to crash.

Exception handling can help deal with runtime errors.

Logic Errors

Section 5: Structure of a C++ Program

- Secion Overview
- Overview of the Structure of a C++ program
- #include Preprocessor directive
- Comments
- The main() function
- Namespaces
- Basic Input and Output (I/O) using cin and cout

Overview of the Structure of a C++ program

#include Preprocessor directive

The c++ preprocessor is a program that process your source code before the compiler sees it.

The c++ processor first strips all the comments from the source file and replaces each comment with single space and then it look for preprocessor directives and executes them.

Pre-processor directives are lines in the source code that begin with a pound or # (hash) symbol.

The most commonly used pre-processor directive is the include directive.

When the pre-processor sees this directive , it replaces the pound include line with the file that it is referring to that it recursively process that file as well.

So by the time compiler sees the source code, all comments are stripped out and all preprocessor directives have been processed and removed.

Preprocessor directives have commonly used to conditionally compile the code.

The C++ preprocessor does not understand the C++.

It simply follows the preprocessor directives and gets the source code ready for the compiler.

The compiler is the program that does understand c++.

Comments

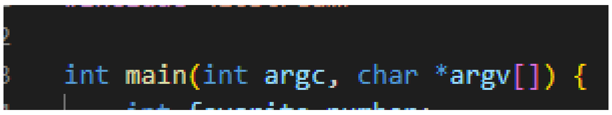
Single line comment //

Multi line commnets

/*

*/

The main() function



```
int main(int argc, char *argv[]) {
```

argc: argument count

To pass arguments from command line.

argv: array of strings

Namespaces

Namespaces are used to reduce the possibility of naming conflicts.

The double colon operator is called the scope resolution operator.

It used to resolve which name you want to use.

using namepace directive

using namespace std;

This is not a better solution for large programs.

using namespace in the programs brings all the names that are defined in the namespace.

So the possibility exists that there will be a still naming conflicts.

C++ provides another variant of using namespace directive.

using std::cout; // use only what you need using std::cin;

using std::endl;

Basic Input and Output (I/O) using cin and cout

cout, cin, cerr and clog are objects representing streams.

cout: cout is an output stream that defaults to the console or the screen.

cin:

it is an input stream that defaults to the keyboard.

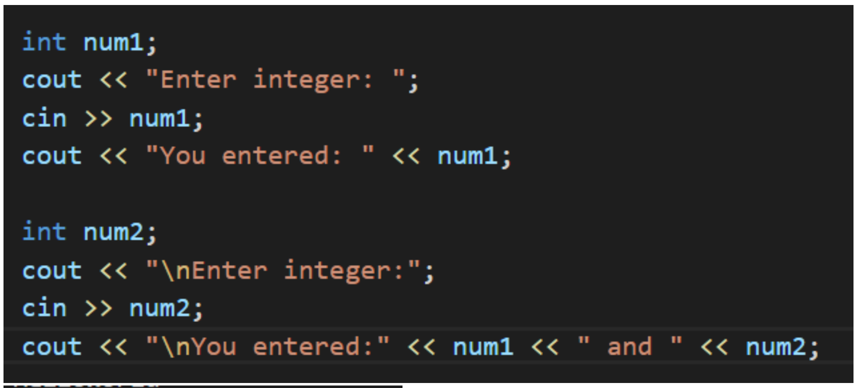
cin extraction uses white space such as spaces, tabs, new lines as terminating the value being extracted.

if you put the spaces between the things that you type in, the spaces will be ignored

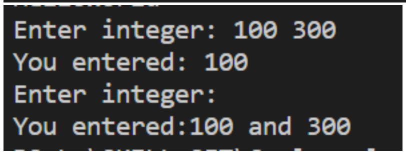
<<: insertion operator

>>: extraction operator

Example 1:



```cpp
int num1;
cout << "Enter integer: ";
cin >> num1;
cout << "You entered: " << num1;

int num2;
cout << "\nEnter integer:";
cin >> num2;
cout << "\nYou entered:" << num1 << " and " << num2;
```

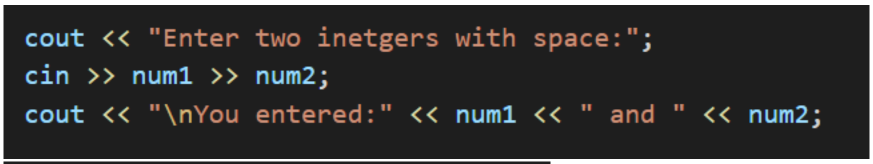


```
Enter integer: 100 300
You entered: 100
Enter integer:
You entered:100 and 300
```

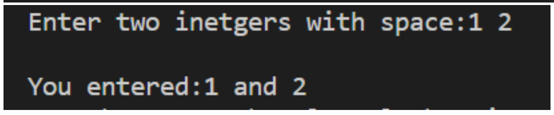
Buffer

```
100 300
```

Example 2:

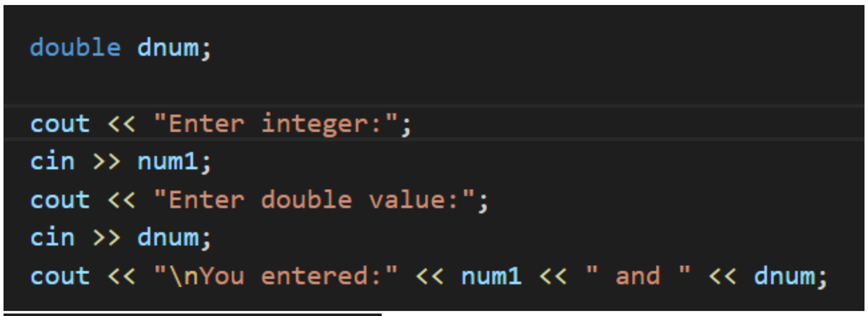


```
cout << "Enter two inetgers with space:";
cin >> num1 >> num2;
cout << "\nYou entered:" << num1 << " and " << num2;
```

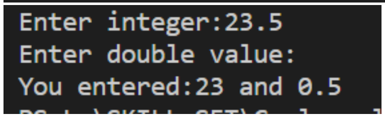


```
Enter two inetgers with space:1 2

You entered:1 and 2
```

Example 3:



```
double dnum;

cout << "Enter integer:";
cin >> num1;
cout << "Enter double value:";
cin >> dnum;
cout << "\nYou entered:" << num1 << " and " << dnum;
```



```
Enter integer:23.5
Enter double value:
You entered:23 and 0.5
```

Buffer

```
23.5
```

23 is integer

0.5 is decimal value

Section 6: Variables and Constants

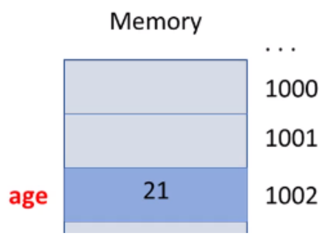
- What is variable?
- Declaring and Initializing variables
- Global variables
- C++ Built-in Primitive Types
- Wht is the size of a variable (sizeof)
- What is constant?
- Declaring and Using Constants

What is variable?

Random Acess Memory (RAM) is a contiguous block of storage used by the computer to store information.

This information includes computer instructions as well as data.

Many programming languages allows us associate a name to memory location.



In above image, the memory location 1002 has been associtaed with name “age”.

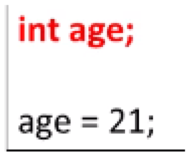
In computing this is called a binding.

A variable is an abstraction for memory location.

Allow programmers to use meaningful names and not memory addresses.

Variables have

- Type – their category (integer, real number, string, Person, Account…)
- Value – the contents (10. 3.14. “Frank”…)



```
int age;

age = 21;
```

In the code snippet, we first declare age to be an integer.

Now the compiler knows that only intgers are allowed to store in age. This is called static typing is because the compiler is enforcing these rules when the programs is compiled

rather than when the program is executing.

Variables must be declared before they are used.

A variables value may change.

Declaring and Initializing Variables

Declaring Vaiables

VaraibleType VariableName;

int age;

double rate;

string name;

Account franks_account;

Person james;

Naming Variables:

Can contain letters, numbers, and underscores.

Must beging with a letter or (_) underscore

- cannot beging with a number

Cannot use c++ reserved keywords

Cannot redeclare a name in the same scope

- remember that c++ is case sensitive

Naming Varibles – Styles and Best Practices

Be consistent with your naming conventions

- myVariableName vs my_variable_name
- avoid beginning names with underscores

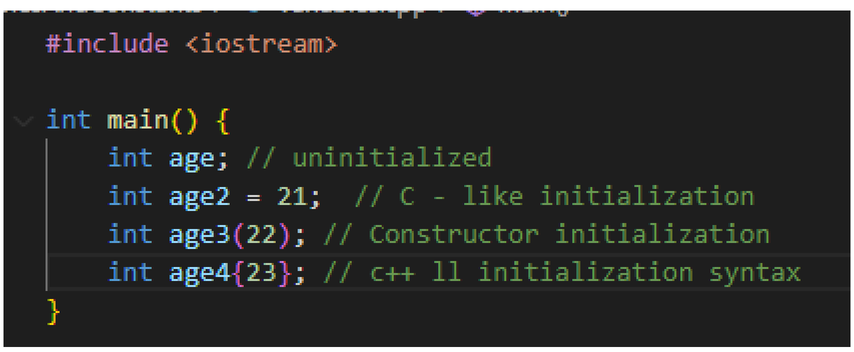
Use meaningful names

- not too long and not too short

Never use variables before initializing them.

Declare variables close to when you need them in your code.

Initializing Variables



```
#include <iostream>

int main() {
    int age; // uninitialized
    int age2 = 21;  // C - like initialization
    int age3(22); // Constructor initialization
    int age4{23}; // c++ 11 initialization syntax
}
```

Global Variables

C++ Built-in Primitive Types

What is the Sie of the variables(sizeof)

What is a Constant?

Declaring and Using Constants