CS 31: Discussion 1C

ABDULLAH-AL-ZUBAER IMRAN

WEEK 1: INTRODUCTION AND BASICS

Welcome!

Email: zubaerimran07cs@cs.ucla.edu

Office Hours: Tuesdays 8:30 am-11:30 am

Location: Boelter 3256S (in the back)

Discussion Objectives

Review and practice things covered during lectures

- Brief review
- Coding examples

Clarifications for project assignments

- Writing real codes in C++
- Getting prepared for the project assignment

Time for you to ask questions!

Lecturing OR Problem Solving?

Less lecturing, more problem solving

30 % Lecture

70% Problem Solving

Outline

Programming Language Intro

Review

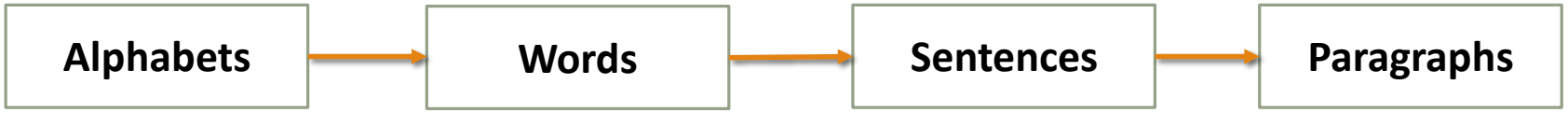
Project 1

- Environment setup
- Compile and debug

Computer Program

What is a computer program?

- A computer program is just a collection of the instructions necessary to solve a specific problem.
- Analogy between learning English language and C++?
  - Steps in learning English



  - Steps in learning C++?

Examples?

- Basic operations– Instruction set
- Approach or method—Algorithm
- Example?
  - Adding two numbers
    - Instruction set→ input, addition, print
    - Algorithm→1. Take the first number
      2. Take the second number
      3. Add two numbers
      4. Print the sum

Review

A computer program is written by a computer programmer in a programming language.

- E.g.: C++, C, java, etc.

What is a programming language?

	Human Language	Programming Language	Machine Language
	English Spanish Italian ...	C C++ Java ...	binary numbers
for humans	easy	medium	difficult
for computers	difficult	medium	easy

It is a language of “medium difficulty” for both us and computers. We use it to represent the logic of a program.

Review

Example:

In English:

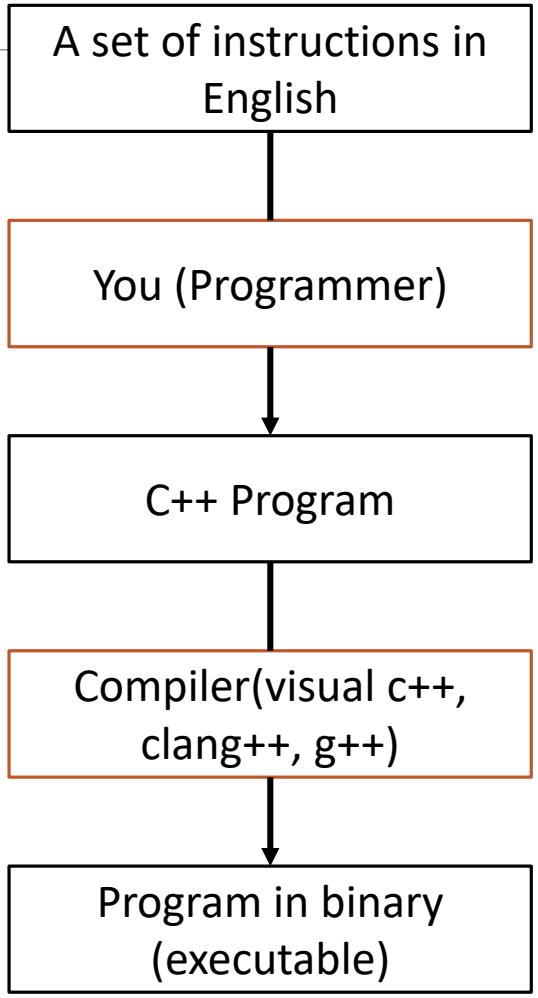
- Hey Computer, say “Hello!”

In a programming language (C++):

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "Hello!" << endl;
6 }
```

In a machine language:

- 010000101110010…



Our first program

```
#include <iostream>
```

Include the library <iostream> to use “cout”

```
using namespace std;
```

Use the namespace std (standard)

namespace is a collection of name definitions

a function name can have different definitions in two namespaces

```
int main()
```

Main() function: where the C++ program begins executing

```
{
        cout << “Hello!” << endl;
}
```

An instruction to print “Hello” in the screen.

endl -> to end the line

Let’s try to compile it!

Compiler:

Source program→COMPILER →Object program

Integrated Development Environments (IDE):

Allows us to easily manage programs, edit files, compile, link, run, and debug programs.

Clang++(Xcode)

Visual C++

g++:

- g31 -o hello hello.cpp

Variables

```
/*
   A simple program demonstrating the usage of variables.
*/

#include <iostream>

using namespace std;

int main()
{
   int x;   // create an integer variable x
   int y;   // create an integer variable y
   x = 5;   // store 5 in x
   y = 6;   // store 6 in y
   cout << "x = " << x << ", y = " << y << endl;   // print them out
   return 0;
}
```

Output: x = 5, y = 6

`int x;` Declare a variable

datatype identifier

int: integer, 4 bytes long, ranging from -2,147,483,647 to 2,147,483,647

Double: real numbers, 8 bytes long, ranging from -1.7×10^{308} to 1.7×10^{308}

Variables

Identifiers

- There are variable naming rules. An identifier must begin with an **alphabetic character (a-z or A-Z) or an underscore ('_')**, which may be followed by **alphabetic/numeric (0-9) characters and underscores**.
- Violating any of these rules will result in a compile error.

Question:

Which ones of these are valid identifiers?

los angeles computer_science engineering

bruin@ucla.edu C++ 5272BH cs31

Variables

Identifiers

- Case sensitivity

Question:

Do $\mathrm{cs31}$ and $\mathrm{CS31}$ refer to the same variable?

Comments

```
/*
   A simple program demonstrating the usage of variables.
*/

#include <iostream>

using namespace std;

int main()
{
   int x;   // create an integer variable x
   int y;   // create an integer variable y
   x = 5;   // store 5 in x
   y = 6;   // store 6 in y
   cout << "x = " << x << ", y = " << y << endl;   // print them out
   return 0;
}
```

We use /* ... */ and // for making notes in the code for ourselves. Text within /* and */ (which can span over multiple lines), and everything after // and before the end of the line are not considered to be part of the code. We call such text a **comment** and say that part of the code is **commented out**.

Errors/Bugs

Compilation/Syntax Errors

- Errors in which the programmer has violated a portion of the language syntax (the language structure). These will prevent the code from compiling.
- Some common syntax errors:
  - Missing semicolons at ends of statements
  - Missing brackets around blocks
  - Missing namespace or #include definitions
  - Misspelled variables or names

Errors/Bugs

Runtime/logic errors:

- Errors that might compile successfully, but encounter an error during runtime that either causes the program to break or produces unexpected (read: wrong) results.
- Some common runtime errors:
  - Division by 0
  - Overflow (e.g.: trying to hold a really big number in an int variable that exceeds its bounds)

Errors/Bugs

Question:

Find the error in the following code snippet. Is it an syntax or a runtime error?

```
int main() {
        int age = 32;
        double  weight = 56;
        cout << “age: ” << Age << “ weight: ” << weight << endl;
}
```

Errors/Bugs

Question:

What could possibly go wrong during runtime with the following program?

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

int main() {
        double numerator, divisor, result;
        cout << “Please enter the numerator: ”;
        cin >> numerator;
        cout << “Please enter the divisor: ”;
        cin >> divisor;
        result = numerator / divisor;
        cout << “The quotient is: ” << result << endl;
}
```

Errors/Bugs

Question:

Will the following code compile? If so, what value will be printed?

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

int main() {
	int x;
	cout << x << endl;
}
```

When a primitive variable is not initialized (i.e., set to some value like int x = 5;), it will have unpredictable junk value.

Errors/Bugs

Question:

Will the following code compile? If so, what value will be printed?

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

int main() {
	double i_declare_war-or-just_this_var = 1;
	cout << i_declare_war-or-just_this_var << endl;
}
```

Errors/Bugs

Question:

Will the following code compile? If so, what value will be printed?

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

int main () {
	int numerator = 5;
	int divisor = 2;
	int result = numerator / divisor;
	cout << result << endl;
}
```

Project 1

One difficult point

- In step 5, find input integer values that cause it to produce incorrect, unusual, or nonsensical results. (Notice that the instructions say to use input integer values, not input like 12324.435 or “computer”.)
- Normal results: 0 ~ 100%
- Incorrect results: -19.5%, 1200%, two percentages whose sum is not 100%, …

Project 1

The zip file you submit must follow the instructions exactly. (Pay attention to the name of each cpp file and your zip file)

Zip file must include: original.cpp, logic_erro.cpp, compile_error.cpp, and report.doc/report.docx/report.txt

Include your name and UCLA ID on the report.

Your code should run successfully under two compilers: g++ with linux and either Visual C++ or clang++ (Xcode).

Due at 11:00 PM Tuesday, April 9
Late submissions will be penalized.

Discussion Slides

You can access today's discussion slides at

All the discussion materials will be available at

https://github.com/zubaerimran/S19-CS31-1G

*Some of the materials presented have been taken from previous TA discussions