Week 1: Hello World!

Muhao Chen
• Teaching Fellow
• Email address: muhaochen@ucla.edu
• Office Hours:
  • Thursday 11:30 ~ 2:30 PM  BH2432
• Personal office
  • BH3551
• Homepage (where I post slides):
  • http://yellowstone.cs.ucla.edu/~muhao/
Muhao Chen

• Ph.D. Candidate in CS, UCLA (w/ Prof. Carlo Zaniolo) 2014-present
• B.S. in CS, Fudan Univ. 2014

Research interest
Neural language models, deep learning for NLP, knowledge bases

This is a picture of me unsuccessfully skiing.
About CS31

• Skills for programming using C++ (without data structures)
• Basic data type models
• Basic principles of memory allocation
• Basic knowledge of object-oriented programming
Outline

• Review
• How to compile programs
• Project 1
Review

- What is a program?
  - A **sequence of rules and instructions** that describe the logic of specific tasks to be processed by the computer.
    - To calculate some formulas, to train some machine learning models.
    - Operating systems, databases, compilers, network system ...
    - Websites, games ...
What is a programming language?

- A language of medium difficulty to both us and computer. We use it to represent the procedural logic of instructions. Machines map it to machine language and execute the instructions.
• Example
• In Human language:
  • Print out “Hello World!” on the screen.

• In a programming language:
  • cout << “Hello World!” << endl;

• In machine language:
  • 0101101111010101010110101010 ...
A punch card used in 1960s to program Fortran (the very early programming language which was extremely close to machine languages)
Compile a Program

#include <iostream>
using namespace std;

int main()
{
    int a = 1, b = 1;
    cout << a + b << endl;
}

• Include the <iostream> library to use “cout”
• Use namespace std (standard)
  • Namespace is a collection of name definitions
  • A function name can be given different definitions in two namespaces
• endl – output a new line
• main() function: where the C++ program begins its logic.
• Note: it is case-sensitive in C++
Compilers

• Compiling with a Visual Studio (VC++)
  • Wysiwyg
  • ** (Choose win32/win64 console application when you create the project!)

• Compiling with g++
  • `g++ -g source_code.cpp -o target`
  • `./target`
Errors

• What is a compile error?
  • Fails to compile.
  • Syntax errors, library errors, link errors, etc.

• What is a logical error?
  • Compiles successfully.
  • Program may run well / Or may crash (e.g. infinite loop, over-allocated memory, etc).
  • Gives incorrect results / undefined behaviors.
```cpp
int main()
{
    int numSurveyed;
    int numApprove;
    int numDisapprove;

    cout << "How many people were surveyed? ";
    cin >> numSurveyed;
    cout << "How many of them approve of the way the president is handling his job? ";
    cin >> numApprove;
    cout << "How many of them disapprove of the way the president is handling his job? ";
    cin >> numDisapprove;

    double pctApprove = 100.0 * numApprove / numSurveyed;
    double pctDisapprove = 100.0 * numDisapprove / numSurveyed;

    cout.setf(ios::fixed); // see pp. 32-33 in Savitch 6/e
    cout.precision(1);

    cout << endl;
    cout << pctApprove << "% say they approve." << endl;
    cout << pctDisapprove << "% say they disapprove." << endl;

    if (numApprove > numDisapprove)
        cout << "More people approve than disapprove." << endl;
    else
        cout << "More people disapprove than approve." << endl;
}
```
Project 1

• One thing we should pay attention to
  • In step 5, find input integer values that cause it to produce incorrect, unusual, or nonsensical results.
  • Is this to cause a compile error or a logical error?
  • Note: The variables numSurveyed, numApprove, and numDisapprove are integer types, so it is not the case to input floating type values like 12.3456.
  • What values should we input to trigger incorrect results?
  • Incorrect results: numSurveyed != numApprove + numDisapprove (e.g. 3000, 2000, 2000)

Unusual or nonsensical results?
### Data types

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>int (Integer)</td>
<td>4Bytes (=32bits) or 8Bytes (=64bits)</td>
</tr>
<tr>
<td>double (double precision float)</td>
<td>8Bytes (=64bits)</td>
</tr>
<tr>
<td>float (single precision float)</td>
<td>4Bytes (=32bits)</td>
</tr>
</tbody>
</table>

- long int, unsigned int, char, boolean …
int (suppose it’s on a 32bits system)

• int: 4Bytes. Range: -2147483648~2147483647 (-2^{31}~2^{31}-1)

5 =

\[
\begin{array}{c}
0000000000000000000000000101
\end{array}
\]

Signed bit (S)  Value bits (V)

-5 =

\[
\begin{array}{c}
1111111111111111111111111011
\end{array}
\]

If S = 0: Value = V
If S = 1. Value = V - 2^{31}

int a=2147483647;  // 0 1111 ... 1111, i.e. 2^{31} - 1
a += 1;

/// How much is a now?
Bit overflow

- A bit overflow occurs when an arithmetic operation attempts to create a numeric value that is too large to be represented within the available storage space.
- So just assign 999999999999999999999999999999999999999999999 also gives logic error.
Project 1

• The zip file you submit must follow the instructions **exactly**. (Pay attention to how to name each cpp file and your zip file!)
• Be careful about compile error and logical error.
• Projects submitted after the due time will receive **reduced or no credit**.

• (Something about David’s projects)
Next week

• Data types and variables
• Operators
• Conditions
• Loops
• I/O
Thank you!