Programming in C++

https://fan1x.github.io/cpp21.html
tomas.faltin@matfyz.cuni.cz

Basic information

- Email: tomas.faltin@matfyz.cuni.cz
- Lab's web: https://fan1x.github.io/cpp21.html
- ZOOM for distance learning
  - https://cuni-cz.zoom.us/j/94350923737
  - Credentials in SIS/mail
- Mattermost
  - Invite link: https://ulita.ms.mff.cuni.cz/mattermost/signup_user_complete/?id=z1knw5ag6p8nipop1i7iciga6a
    - Use ASAP, might expire eventually
  - Channel: `nprg041-cpp-english`
- Gitlab
  - https://gitlab.mff.cuni.cz/
  - https://gitlab.mff.cuni.cz/teaching/nprg041/2021-22/eng

Communication is the key

- Don’t be afraid to ask
  - via email
  - on Mattermost (instant)
    - DM if related to you only
    - Into a channel if others can benefit from it
- If you struggle with something
- If you feel like you might miss a deadline
- Be proactive

Labs credit

- Submitted homeworks before Monday midnight (to Gitlab)
  - Even if not attending!
  - Won’t be graded, for a feedback
- Two large homeworks in ReCodex (40 points)
  - Points are included in the final score from the course
  - Smaller HW – 15 points, ~November
  - Larger HW – 25 points, ~December
- Software project
  - Topic must be approved by 28/11/2021
  - First submission: 24/4/2022
  - Final submission: 22/5/2022
  - **All the steps typically mean multiple iterations within multiple days. If you wait for the last minute, there is a chance you won’t make it**

Code Requirements

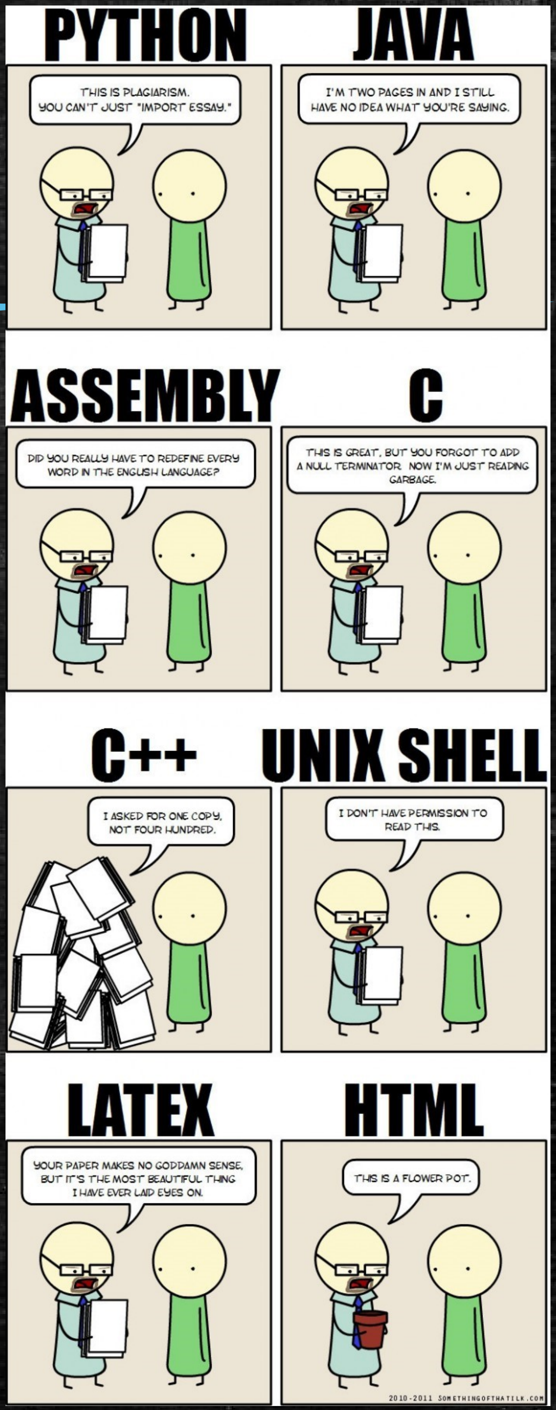
- Consistency
  - Be consistent within the code – keep a single code style
- Cleanness, readability
  - Code doesn’t contain commented/dead parts
  - Code should be readable on its own
- Safe, modern
  - E.g., prefer `std::vector<int>` to `new int[]`
- Working
  - OFC, if the code is not working, all the above points are not that important, but they will help you with debugging at least ☺

Why C++

"C makes it easy to shoot yourself in the foot. C++ makes it harder, but when you do, it blows away your whole leg."
-- Bjarne Stroustrup

"It was only supposed to be a joke, I never thought people would take the book seriously. Anyone with half a brain can see that object-oriented programming is counter-intuitive, illogical and inefficient."
-- Stroustrup C++ 'interview' (https://www-users.cs.york.ac.uk/susan/joke/cpp.htm)

C++ != speed, C++ ~ control


PYTHON
THIS IS PLAGIARISM. YOU CAN'T JUST "IMPORT ESSAY."
JAVA
I'M TWO PAGES IN AND I STILL HAVE NO IDEA WHAT YOU'RE SAYING.
ASSEMBLY
DID YOU REALLY HAVE TO REDEFINE EVERY WORD IN THE ENGLISH LANGUAGE?
C
THIS IS GREAT, BUT YOU FORGOT TO ADD A NULL TERMINATOR. NOW I'M JUST READING GARBAGE.
C++
I ASKED FOR ONE COPY, NOT FOUR HUNDRED.
UNIX SHELL
I DON'T HAVE PERMISSION TO READ THIS.
LATEX
YOUR PAPER MAKES NO GODDAMN SENSE, BUT IT'S THE MOST BEAUTIFUL THING I HAVE EVER LAID EYES ON.
HTML
THIS IS A FLOWER POT.
2010-2011 SOMETHINGOFTHATILK.COM

Working Environment

- Use anything you like ☺
- IDEs
  - Visual Studio
    - License for students at https://portal.azure.com/…
  - VS Code
  - Clion
  - Code::Blocks
  - Eclipse
  - …
- Compilers
  - MSVC, GCC, Clang+LLVM, ICC, …

C++ (interesting) links

- Reddit, Slack, …
- https://en.cppreference.com/w/
- http://www.cplusplus.com/
- http://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines
- https://www.youtube.com/user/CppCon
- https://isocpp.org/
- http://www.open-std.org/jtc1/sc22/wg21/docs/papers/
- https://gcc.godbolt.org/
- …

Hello World

```
#include <iostream>
#include <string>

int main() {
  std::string name;
  std::cin >> name;
  std::cout << “Greetings from ” << name << std::endl;
  return 0;
}
```

Hello Worl

```
#include <iost
#include <str

int main() {
   std::string name;
   std::cin >> name;
   std::cout << “Greetings from ” << name << std::endl;
   return 0;
}
```

Include the libraries which implements the used STL constructs (string, cin, cout)

The main entry point/function for all programs. The execution starts here

Read from standard input (keyboard)

Write to standard output (screen)

All the STL constructs live inside `std` namespace

More Complex Program

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

int length(const string& s) { ... }

void pretty_print(const vector<string>& a) {   ... a[i] ...  }

int main(int argc, char** argv) {
  vector<string> arg(argv, argv+argc);
  if (arg.size() > 1 && arg[1] == "--help")  {
    cout << "Usage: myprg [OPT]... [FILE]..." << endl;
    return 8;
  }
  pretty_print(arg);
  return 0;
}
```

More Complex Program

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

int length(const string& s) { ... }

void pretty_print(const vector<string>& a)          [i] ...   }

int main(int argc, char** argv) {
  vector<string> arg(argv, argv+argc);
  if (arg.size() > 1 && arg[1] == "--hel
    cout << "Usage: myprg [OPT] .. [FILE]..                ;
    return 8;
  }
  pretty_print(arg);
  return 0;
}
```

Include the whole std namespace

Passing the argument by (const) reference

Arguments of the program on the command line

Transform the arguments into C++ array of strings

Homeworks

1. Hello World
2. A greeting program (use names from arguments)
   - `hello.exe Adam Eve` → `Hello to Adam and Eve`
   - What is inside `args[0]`?
3. Summation of numbers from arguments
   - `sum.exe 1 2 3 4 5` → `15`
   - `stoi(), stod(), stoX()`
     - Functions for transformation from **s**tring **to** <something>
4. A simple calculator (only for operations +-)
   - `calc.exe 1+2+3-4` → `2`
   - to Gitlab
   - The previous programs are not needed, they should give you a lead