Introduction to Programming

Week 1

Introduction

Anatomy of your first C++ program

includes I/O library →

main() method →

body of main() method →

```
#include<iostream>

int main()
{
  std::cout << “Hello World”;

}
```

Anatomy of your next several programs

includes I/O library →

main() method →

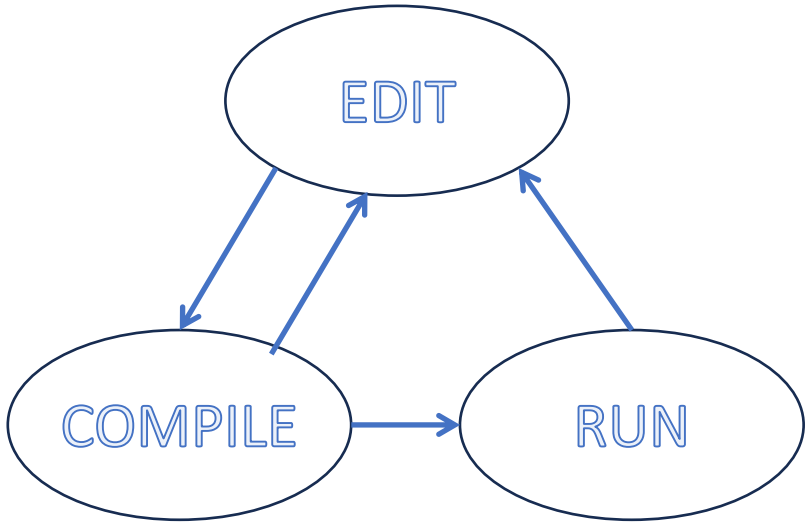
body of main() method →
(a sequence of statements)

```
#include<iostream>

int main()
{
    …

}
```

Program development in C++

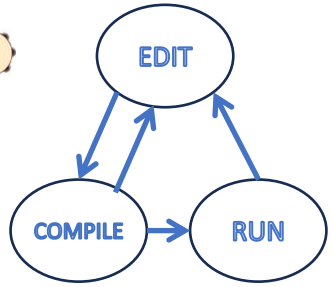
A three-step process, with feedback

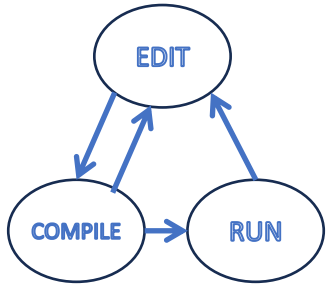
1. **EDIT** your program
   - Create it by typing on your computer's keyboard.
   - Result: a text file such as HellpWorld.cpp
2. **COMPILE** it to create an executable file
   - Use the C++ compiler.
   - Result: An executable file.
   - Mistake? Go back to 1. to fix and recompile.
3. **RUN** your program
   - Result: Your program's output.
   - Mistake? Go back to 1. to fix, recompile, and run.



Software for program development

Any creative process requires cyclic refinement/development





A significant difference: *we can use software to facilitate the process*

Software development environment: Software for editing, compiling, and running programs.

Two time-tested options

- Virtual terminal
- Integrated development environment (IDE)

Software for program development: tradeoffs

Virtual Terminal

Pros

- Approach works with any language.
- Useful beyond programming.
- Used by professionals.
- Has withstood the test of time.

Cons

- Good enough for long programs?
- Dealing with independent applications.
- Working at too low a level?

IDE

Pros

- Easy-to-use language-specific tools.
- System-independent (in principle).
- Used by professionals.
- Can be helpful to beginners.

Cons

- Overkill for short programs?
- Big application to learn and maintain.
- Often language- or system-specific.