Outline

• Introduction to computing
• Some simple programs
• Remarks on programming
• Spirit of the course
Computers are everywhere!

- Cell phones, tablets, desktops, game consoles, e-book readers, cameras, televisions contain a computer
- Computers used to:
  - Book train/plane/bus tickets
  - Search the internet
  - Predict the weather
  - ...
- **Goal of the course:** Learn how to make computers do things such as the above
A computer is a giant electrical circuit that can

- Receive data from the external world
  - data = numbers,
  - images, sounds can also be represented using numbers and hence fed to a computer
- Perform calculations on the data it receives
- Send the results back to the external world
- What calculations to perform: determined by a program that must be loaded in the computer
Programs

- **Program** = a precise description of the calculations we want the computer to perform
- By feeding different programs to a computer you can make it do different calculations.
- This course tells you how to construct ("write") programs.
- Special notation is to be used to write programs: "Programming Language"
The C++ programming language

• Designed by Bjarne Stroustrup, 1980s. Derived from the C programming language.
• Substantial evolution. Still continues.
• Early part of the course: C++ augmented with a package called simplecpp
• Simplecpp: More fun and easier to use than bare C++. Built-in graphics.
Goal for today

- Write some small programs using C++
- The programs will draw pictures on the screen.
- Use “Turtle Simulator” contained in simplecpp
  - Based on Logo: A language invented for teaching programming by Seymour Pappert et al.
- We “drive” a “turtle” on the screen!
- To drive the turtle you write a C++ program.
- Turtle has a pen, so it draws as it moves.
- Drawing pictures may seem too much fun, but if you master it, you have mastered a lot of programming
#include <simplecpp>

main_program{
    turtleSim();
    forward(10); right(90);
    forward(10); right(90);
    forward(10); right(90);
    forward(10);
    wait(5);
}

• “Use simplecpp facilities”
• Main program begins
• Use turtle simulator
• forward(n): “Move the turtle n pixels in the direction it is currently headed.”
• right(d): “Make the turtle d degrees to the right.”
• wait(t): “Do nothing for t seconds.”
• End of main program
How to run this program

• Install simplecpp on your computer, by itself or using the codeblocks IDE.
• Type in the program into a file. Call it turtle1.cpp
• “Compile” it:
  – On unix run: s++ turtle1.cpp
  – On code blocks: use compile button
• Execute it:
  – On unix, run: ./a.out
  – On code blocks: use run button
General Ideas

• C++ program = sequence of commands.
  ```
  main_program { ... written here ... }
  ```
• Statement/command: terminated by “;”
• Arguments: additional data needed by command to do its work.
  – forward: how much forward?
  – right: what angle?
  – () if no arguments, e.g. turtleSim()
General Ideas (contd)

• Commands are executed from top to bottom, left to right.
  – Usually..

• “Compiling” the program: translating it into a form that your computer can “understand”.
  – More on this in chapter 2.

• Executable file: result of compilation. On unix the file has the name a.out by default.