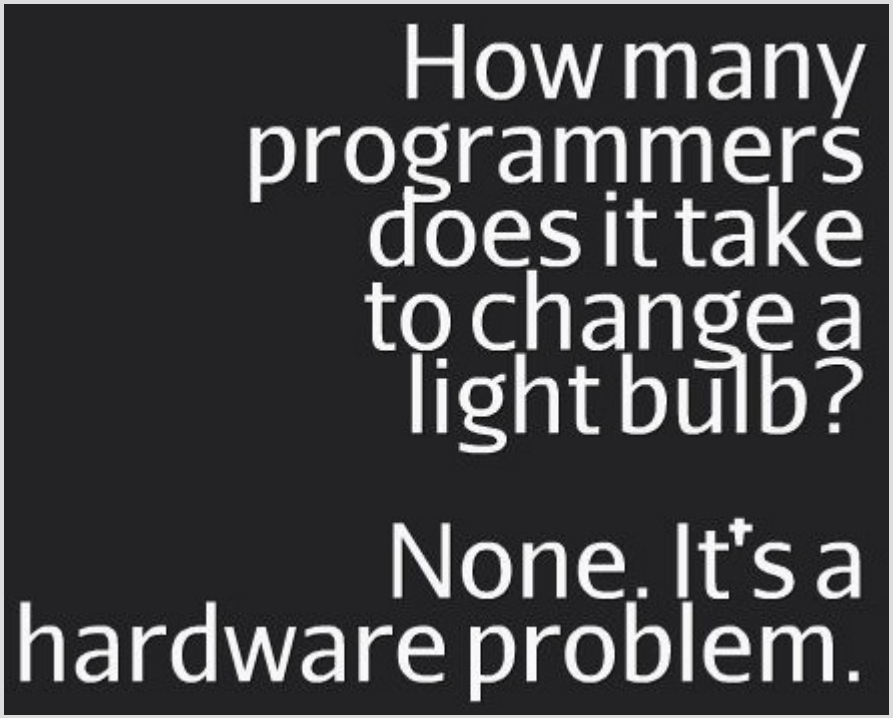
Introduction

Ch 1



Object Oriented

Main focus is on objects/variables and how they interact (represented by me as boxes)

Reusable groups of actions (verbs) between objects are called functions (squiggly boxes)

These actions can take additional information called arguments,
(an analogy is ordering at a restaurant; the ordering format is the same, different food)

Object Oriented

One format is:
object.function(argument, argument...);

Example:
James.teaches(CSci 1113);
teach(James, CSci 1113);

The dot (period) shows that “teaching” is an action done by “James”

Banana Nut Bread

Ingredients

* 3 or 4 ripe bananas
* 1/3 cup melted butter
* 1 cup sugar
* 1 egg, beaten
* 1 teaspoon vanilla
* 1 teaspoon baking soda
* Pinch of salt
* 1 1/2 cups of all-purpose flour
* 1 cup of nuts

Data (Objects)

Banana Nut Bread

Directions

1. Preheat the oven to 350°F (175°C).
2. Mix butter into the mashed bananas in a large mixing bowl.
3. Mix in the sugar, egg, and vanilla.
4. Sprinkle the baking soda and salt over the mixture and mix in.
5. Add the flour and nuts last, mix.
6. Pour mixture into a buttered 4x8 inch loaf pan.
7. Bake for 1 hour. Cool on a rack.

Banana Nut Bread

Directions

1. Preheat the oven to 350°F (175°C).
2. Mix butter into the mashed bananas in a large mixing bowl.
3. Mix in the sugar, egg, and vanilla.
4. Sprinkle the baking soda and salt over the mixture and mix in.
5. Add the flour and nuts last, mix.
6. Pour mixture into a buttered 4x8 inch loaf pan.
7. Bake for 1 hour. Cool on a rack.

Banana Nut Bread

Directions

1. Preheat the oven to 350°F (175°C).
2. Mix butter into the mashed bananas in a large mixing bowl.
3. Mix in the sugar, egg, and vanilla.
4. Sprinkle the baking soda and salt over the mixture and mix in.
5. Add the flour and nuts last, mix.
6. Pour mixture into a buttered 4x8 inch loaf pan.
7. Bake for 1 hour. Cool on a rack.

Banana Nut Bread

Pseudo code directions

1. oven.preheat(350);
2. bowl.mix(butter, bananas);
3. bowl.mix(sugar, egg, vanilla);
4. bowl.sprinkle(baking soda, salt);
5. bowl.mix(flour, nuts);
6. bowl.pour(pan);
7. pan.bake(60);
8. pan.cool();

Banana Nut Bread

Pseudo code directions #2

1. oven.preheat(350);
2. bowl.add(butter, bananas);
3. bowl.mix();
4. bowl.add(sugar, egg, vanilla);
5. bowl.mix();
6. bowl.sprinkle(baking soda, salt);
7. bowl.add(flour, nuts);
8. bowl.mix();
9. pan.pour(bowl);
10. pan.bake(60);
11. pan.cool();

Banana Nut Bread

```
mashedBananas = bananas.mashed();
bowl.add(butter, mashedBananas);
```

same as:

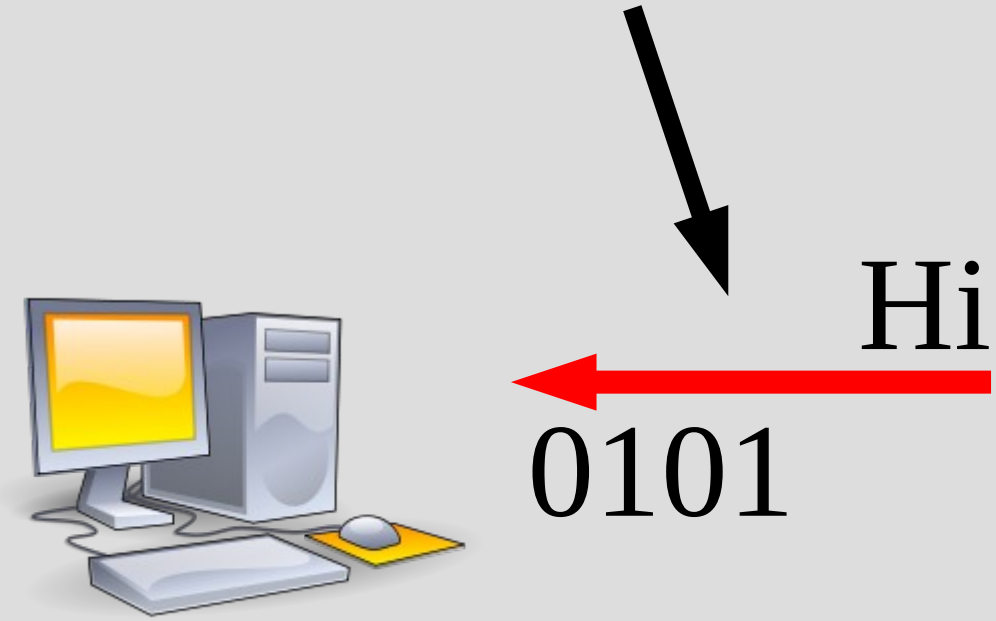
```
bowl.add(butter, bananas.mashed());
```

```
Kitchen.bowl.add(butter, bananas.mashed());
```

```
hand.mix(butter, mashedBananas);
bowl.add(hand.mix(butter, mashedBananas));
```

Compiling

Converting code to binary is called compiling



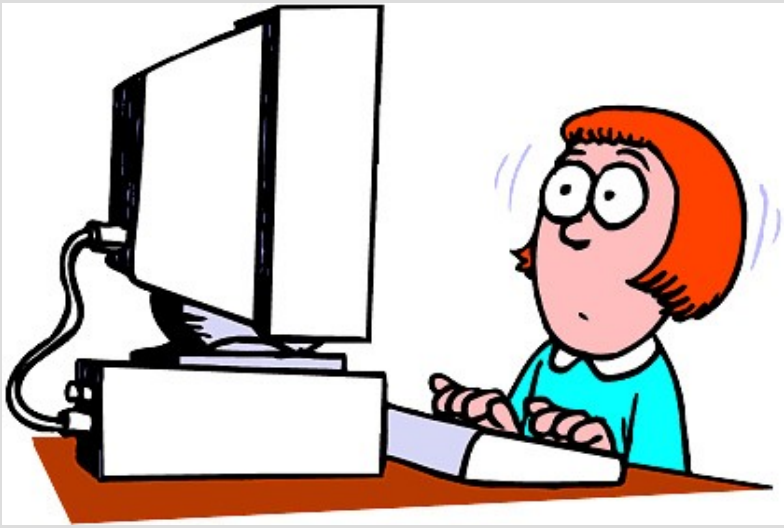
Compiling

Often this compiled code Will not work on other computers

0101

Hi

0101



Compiling


Microsoft
Mac OS
Sie sind nicht berechtigt, unrechtmäßige Kopien dieses Datenträgers zu erstellen.
Sie müssen ein lizenzierter Benutzer von Microsoft Office sein, um die Update-Version installieren zu dürfen.
Microsoft
Office 98
Macintosh Edition
Upgrade
Für Ergebnisse, die so brillant sind wie Ihre Ideen.

Compiling

C++ is a high level language (human readable)

Compiling changes a high level language into a low level language that is easier for the computer (computer cannot run high level)

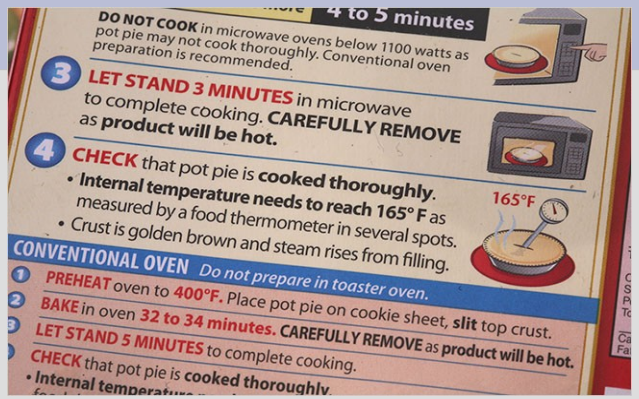
Compiling

Your source code is the original language you wrote your program in (the C++ code for us)

You must recompile the source code every time you save a change before running the program again

Compiling tl;dr

directions



cook

meal

eat

satiated

code

compile

1's and 0's (program)

run

pretty colors

Compiling

In labs, the computers will come with a program called “geany” (which I will use too)

This program is where you can write code and easily compile simple programs

To run it either click the terminal icon () on the left bar or press Ctrl+Alt+T

Then type: geany (enter)

High level (C++)

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

int main ()
{
  cout << "Hello World! ";
  return 0;
}
```

(See: helloWorld.cpp)

Low level (Assembly)

```
MODEL SMALL
IDEAL
STACK 100H

DATASEG
MSG DB 'Hello, World!', 13, '$'

CODESEG
Start:
MOV AX, @data
MOV DS, AX
MOV DX, OFFSET MSG
MOV AH, 09H ; output ascii string
INT 21H
MOV AX, 4C00H
INT 21H
END Start
```

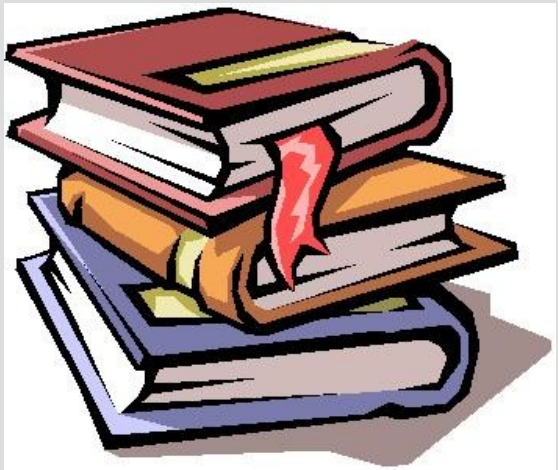
Ease of use

Why C++?

Speed

Control

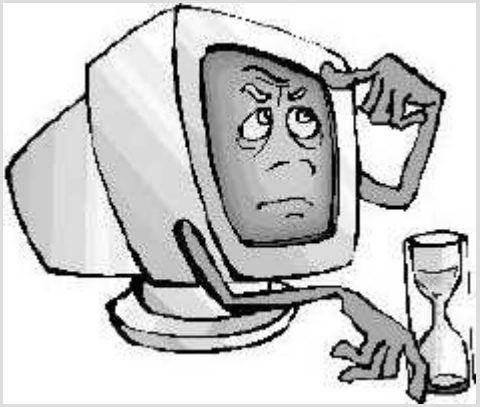
Libraries



Speed

Not all programming languages need to compile code as C++ (Java, Python)

Compiling can greatly increase speed of a program



Control

C++ allows you great control over your data (and its interpretation)

This comes with a burden of responsibility to properly manage your data

If you mismanage your data, you are likely to cause an error in your program

Libraries

C++ is an old language (older than me) and this comes with pros and cons...

Some aspects are quirky to enable backwards compatibility (and are honestly out of date)

Since it has been around for a long time, there are lots of supporting libraries
(and the language continues to develop...)

Java/Python vs C++

Java/Python

Goes anywhere
Comfy

C++

Fast
Fine tuned

Magic 8 ball

Magic 8 ball

What a rip off!

Magic 8 ball

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Maybe.";
7
8      return 0;
9  }
```

Keyboard input

cout << “word”

- prints “word” to the screen

cin >> x

- store what is typed into “x”
  (x is some object or data)

Can also do arithmetic using +, -, / and *
(See: inputOutput.cpp)

Types of errors

Syntax error - code will not compile
e.g. cout(“hi”);

Runtime error - code crashes after starting
(see: runtimeError.cpp)

Logic error - code runs but doesn't return the correct answer
(see: logicError.cpp)

Syntax

Syntax is a fancy word for the “grammar” of programming languages

The basic English syntax is:
(subject) (verb) (noun)
“I eat bananas” not “Bananas I eat”

The computer is VERY picky (and stubborn) about grammar, and will not understand you unless you are absolutely correct!

Avoid errors

To remove your program of bugs, you should try to test your program on a wide range of inputs

Typically it is useful to start with a small piece of code that works and build up rather than trying to program everything and then debug for hours

Comments

Comments are ignored pieces of code (computer will pretend they do not exist)

// denotes a single line that is commented
// (everything before hitting enter)

/* denotes the beginning of a comment and the end of a comment is denoted by */

Additional facts

Braces denote a block of code { }
(belonging to a method, class, etc.)

“White space” is ignored, just as the your brain will ignore the bottom third of this slide (this is why we need a semi-colon)