MAT 685: C++ for Mathematicians

Introduction

John Perry

University of Southern Mississippi

Spring 2017

Outline

1. What this class is about
2. Computer programming
3. Words I shouldn't have to say
4. Summary

Outline

1. What this class is about
2. Computer programming
3. Words I shouldn't have to say
4. Summary

Description

Mathematically-centered introduction to computer programming for mathematicians

Description

Mathematically-centered introduction to computer programming for mathematicians

- *introduction:* there is a huge world beyond this
- *mathematically-centered,* as opposed to *computer science-centered*
    - focus on numbers, computation — not strings, I/O

Outline

1. What this class is about
2. Computer programming
3. Words I shouldn't have to say
4. Summary

Desired outcomes

1. Familiarity with programming
   - transferability to other languages
2. Experience with C++
   - basic syntax
   - preprocessing, compiling, linking, debugging
   - "gcc toolchain"
3. Introduction to design
4. Hands-on experience with external project

Required caveats

Programming like writing proofs

- one class is not enough
- breaking task into pieces
- solving difficult problems
- somewhat steep learning curve
- generalizing and creating abstractions
- advanced problems require advanced techniques
- **cannot learn without *doing* and *struggling***

Required caveats

Programming like writing proofs

- one class is not enough
- breaking task into pieces
- solving difficult problems
- somewhat steep learning curve
- generalizing and creating abstractions
- advanced problems require advanced techniques
- **cannot learn without *doing* and *struggling***

The point

One course will not make you an expert programmer. In fact, it will not even make you a particularly competent one. But you will get out of this what you put into it!

For example

I've been doing this for 30+ years

ca. 1983	BASIC (self-study, unsuccessful)
ca. 1989	Pascal (university: also, advanced concepts)
ca. 1991	C (self-study, internships)
ca. 1999	Java (self-study)
ca. 2003	Eiffel (self-study, research)
ca. 2008	Python (self-study, research)
ca. 2013	C++ (self-study, research)

For example

I've been doing this for 30+ years

ca. 1983	BASIC (self-study, unsuccessful)
ca. 1989	Pascal (university: also, advanced concepts)
ca. 1991	C (self-study, internships)
ca. 1999	Java (self-study)
ca. 2003	Eiffel (self-study, research)
ca. 2008	Python (self-study, research)
ca. 2013	C++ (self-study, research)

The point

- requires self-study
- understanding concepts make self-study possible
- you will probably need this in the future

Why program?

Programming bridges gap between humans, computers

- Control over computer
- Deeper understanding of computer technology

Why program?

Programming bridges gap between humans, computers

- Control over computer
- Deeper understanding of computer technology

- Computers don't understand human languages
  - Humans intuitive, poetic; computers literal, mechanical
  - Computers only understand **on** or **off** ("machine code")

Why program?

Programming bridges gap between humans, computers

- Control over computer
- Deeper understanding of computer technology

- Computers don't understand human languages
  - Humans intuitive, poetic; computers literal, mechanical
  - Computers only understand **on** or **off** ("machine code")

- (Most) humans don't understand computer's language
  - Those that do prefer not to use it
  - Mathematics literal and precise, but (most) humans don't understand it, either!

Programming v. "math systems"

- Pros of programming
  - faster execution time (modulo various caveats)
  - more control
  - more flexibility
    (employment, ability to solve problems)
  - lower cost
    (most compilers free; most math systems $$$)

Programming v. "math systems"

- Pros of programming
  - faster execution time (modulo various caveats)
  - more control
  - more flexibility
    (employment, ability to solve problems)
  - lower cost
    (most compilers free; most math systems $$$)

- Cons of programming
  - longer development time
  - generally inconvenient
  - can be difficult to maintain/extend

Programming v. "math systems"

- Pros of programming
  - faster execution time (modulo various caveats)
  - more control
  - more flexibility
    (employment, ability to solve problems)
  - lower cost
    (most compilers free; most math systems $$$)
- Cons of programming
  - longer development time
  - generally inconvenient
  - can be difficult to maintain/extend
- Can often mix the two

Outline

1. What this class is about
2. Computer programming
3. Words I shouldn’t have to say
4. Summary

Yes, this is a "Dr. Perry" class

Student comments, Fall 2016:

> *If a student went to his office hours for help, he would often tell students to look over something again but not take time the time to explain what he meant or what was said in the book. Sometimes he would explain what he meant but those were few and far between. I would not recommend anyone taking a class from him.*

Yes, this is a “Dr. Perry” class

Student comments, Fall 2016:

> *If a student went to his office hours for help, he would often tell students to look over something again but not take time the time to explain what he meant or what was said in the book. Sometimes he would explain what he meant but those were few and far between. I would not recommend anyone taking a class from him.*

> *Dr. Perry is tougher than the other professors but he teaches every thing we will need in the future where others donot.*

Yes, this is a “Dr. Perry” class

Student comments, Fall 2016:

> *If a student went to his office hours for help, he would often tell students to look over something again but not take time the time to explain what he meant or what was said in the book. Sometimes he would explain what he meant but those were few and far between. I would not recommend anyone taking a class from him.*

> *Dr. Perry is tougher than the other professors but he teaches every thing we will need in the future where others donot.*

(guess which one was written by a math major)

Yes, this is a “Dr. Perry” class

“High” expectations (this is grad school, after all)

- read the text
- do the assigned work
- don’t cheat
- ask questions... but not “dumb” questions

Yes, this is a “Dr. Perry” class

“High” expectations (this is grad school, after all)

- read the text
- do the assigned work
- don't cheat
- ask questions... but not “dumb” questions

Objection: “There's no such thing as a dumb question”

Oh, really?

Yes, this is a “Dr. Perry” class

“High” expectations (this is grad school, after all)

- read the text
- do the assigned work
- don't cheat
- ask questions... but not “dumb” questions

Objection: “There's no such thing as a dumb question”

Oh, really?

- Definitions, grading information, important dates
  - you can look them up (index, syllabus, final exam date)
  - *caveat:* not dumb if I forget to define/assign/record it

Yes, this is a "Dr. Perry" class

"High" expectations (this is grad school, after all)

- read the text
- do the assigned work
- don't cheat
- ask questions... but not "dumb" questions

Objection: "There's no such thing as a dumb question"

Oh, really?

- Definitions, grading information, important dates
  - you can look them up (index, syllabus, final exam date)
  - *caveat:* not dumb if I forget to define/assign/record it
- Any question I've answered 8,587,245 times already
  - turn off phone, log off Facebook
  - **pay attention**

Yes, this is a "Dr. Perry" class

"High" expectations (this is grad school, after all)

- read the text
- do the assigned work
- don't cheat
- ask questions... but not "dumb" questions

Objection: "There's no such thing as a dumb question"

Oh, really?

- Definitions, grading information, important dates
  - you can look them up (index, syllabus, final exam date)
  - *caveat:* not dumb if I forget to define/assign/record it
- Any question I've answered 8,587,245 times already
  - turn off phone, log off Facebook
  - **pay attention**
- A question you could answer yourself in 5 seconds

What is not a dumb question?

Most questions!

- "Can you make the screen text bigger?"
- "Can you go back to the previous slide?"
- "Can I get a hint on this problem?"
- "What you typed isn't working for me. Can you check what I have?"
- "What does this error mean?"
- "Why isn't my program working?"
  (but be ready for questions from me)
- "Will you be in your office today?"
- "What's your website again?"

On the other hand...

The following comments help no one

- "I don't understand."
- "This is complicated."
- "My program isn't working."
- "Other classes don't have to work so hard!"

On the other hand...

The following comments help no one

- "I don't understand."
- "This is complicated."
- "My program isn't working."
- "Other classes don't have to work so hard!"

Common misconceptions

- "I can't do this."
- "I'll never learn."
- "Programming's not for me."

Has cheating been a problem?

Yes

MAT 305

- every one of the last 5 years, probably more
- at least one Honors student involved
- at least one degree denied
- half of a 2016 class was dishonest in one way or another

Has cheating been a problem?

Yes

MAT 305

- every one of the last 5 years, probably more
- at least one Honors student involved
- at least one degree denied
- half of a 2016 class was dishonest in one way or another

The university now has an XF grade.

Has cheating been a problem?

Yes

MAT 305

- every one of the last 5 years, probably more
- at least one Honors student involved
- at least one degree denied
- half of a 2016 class was dishonest in one way or another

The university now has an XF grade.

I am not afraid to use it.

Has cheating been a problem?

Yes

MAT 305

- every one of the last 5 years, probably more
- at least one Honors student involved
- at least one degree denied
- half of a 2016 class was dishonest in one way or another

The university now has an XF grade.

I am not afraid to use it.

You have been warned.
(You are graduate students, so this won't be a problem... *right?*)

"What, me cheat?"

Definitely cheating:

- using someone else's solution to the same problem
- pretty much any online search
  - looking up a solution to a similar problem online
  - adapting an online solution to a similar problem
  - **do not use online help sites**

"What, me cheat?"

Definitely cheating:

- using someone else's solution to the same problem
- pretty much any online search
  - looking up a solution to a similar problem online
  - adapting an online solution to a similar problem
  - **do not use online help sites**

Definitely *not* cheating:

- asking the professor for help
  - I will help you, though it may not be fun
- reading the book
  - adapting a book/class program to a similar problem
- discussing program design/implementaiton with others
  - just don't *copy* or have them tell you what to type

Outline

1. What this class is about
2. Computer programming
3. Words I shouldn't have to say
4. Summary

Summary

- Class on *mathematical* problem solving with *computers*
  - programming via C++
- High expectations
  - hopefully not scared you out of the class