Introduction

1. What this course offers
2. Course topics
3. How the course delivers

Self-introductions

• What are your experiences with computing?
• What do you think CS is about?
• What do you expect from this course?
• What have your learning experiences been like?
• What is your major?
Inquiry

• What words come to your mind when I say “computer program”?
• What is computing?
  • Is digesting food computing?
  • Is digesting an experience computing?
• What do you need to know about computer science and Java programming? As an IT professional; software engineer; CS major

1. What this course offers

• What is computer science? How does it differ from information technology?
• Do you like solving problems?
• What are data and abstraction?
• How does thinking compare with computation?
The chronic software crisis

• The global market demands that new tools be created to work and do business
• New software is required rapidly
• Software development for decades has tended to be behind schedule
• Software is often unreliable
• Solutions: web hosting, design, documentation, readable code, structured techniques, object-oriented technology

Software failures

• Health care site, 2013
• Ariane V rocket explosion
• THERAC injection of radioactive medicine
• $500M airport luggage system
• Any in your experience?
Information technology

- “The set of techniques used in information handling and retrieval of information automatically” (Evans et al)
- Hardware and software for processing digital data (Keil)
- “Application of appropriate technologies to the organization, manipulation, and distribution of information by computers and telecommunications” (G. Stamatellos)

IT fluency and programming

Bundling instructions is programming, e.g.:
- Named styles: bundles of text formats
- HTML files: what is to appear on a web page, including interactive elements
- Spreadsheet formulas: algebraic expressions that use cell references (variables) to compute
- Database queries: select some data from a table
- JavaScript event-handling code in HTML files
- Flowcharts, pseudocode, and UML specify actions
The discipline of computer science

- The study of computer science is concerned with *solving problems about information*
- “The study of natural and artificial information processes” (Peter Denning, 2007)
- The tools include
  - *analysis* (breaking down problems)
  - *algorithms* (step-by-step plans)
  - designs of *interactive processes*
- CS has a *logical* and an *empirical* side
- We use *abstraction* to help solve problems

The CS curriculum

*Threads:*

- **Programming**, system design, and software engineering
- **Computer architecture**, operating systems, and networks
- **Data management** (e.g., databases)
- **Theory** (logic, statistics, discrete math, algorithm analysis, automata, AI)
Cybernetics

• The study of interactive systems driven by feedback
• Norbert Wiener, Cybernetics: Or Control and Communication in the Animal and the Machine, 1948
• Word origin: kyber, person at helm of boat
• Note: Cybernetics was not first defined in relation to computer technology

Three computing paradigms

• Algorithmic

  Finite input → Program → Finite output

• Sequential interactive

  Stream input → Program → Stream output
  Memory

• Multi-stream interactive
Symbol manipulation

- **Symbol**: an abstraction by which we may represent items in the real world
- **Example**: \{0,1\} is a set of symbols
- This set may be used to build symbols of any complexity (numerals, words, pix…)
- We can *operate* on symbols (add, concatenate, reorder, etc.)
- A computation can combine sequences, branches, loops

Some functional problems

- Write an expression whose value is the area of the surface of a 2' x 3' x 5' box
- ... a box \(w\) feet wide, \(h\) feet high, and \(d\) feet deep
- Write an expression whose value is the 6.25% sales tax on a purchase of a $199 hard drive and a $79.95 software package
- ... a hard drive costing \(h\) dollars and software costing \(s\) dollars, taxed at \(t\)%
Computing practices

Categories:
• **Programming** (using programming languages – programs ≠ computer science!)
• **Engineering** systems (hardware and software)
• **Applying** (building systems to support practical work)
• **Innovating** (generating durable changes in how systems and communities operate)
• **Modeling and validation** (to represent systems abstractly; to predict and verify behavior)

Guiding concerns in CS I

• Algorithm and module design
• Syntax, semantics of programming languages
• Documentation
• Objects and classes of objects

*Steps in problem solving*

*Repeat until problem is solved:*
1. Specify the desired results
2. Design a solution
3. Code solution in programming language
4. Compile, test, debug
The system development process

**Phases (repeated if necessary):**

- **Analysis:** specifies input, output
- **Program design:** prepares algorithms, data structures
- **Coding:** implements design as a program in a language
- **Testing** of working program
- **Maintenance:** addresses errors and needs not found in previous stages

---

Testing and debugging

- Software and web sites require testing before deployment
- Testing is often done by quality assurance departments
- All software writing entails error and debugging
- JavaScript is easy to test on a browser, but the browser does not supply error locations or other diagnostics
Kinds of *abstractions* used in CS I

- *Bits* store data of any type -- numeric, text, sound, image
- *Variables* and other expressions with values; variables occupy storage
- *Data types* (string, integer, real)
- *Control structures* (e.g., loop, branch)
- *Algorithms and interactive process specifications*
- *Classes*: specify categories of objects
- *Objects*: instances of classes

CS I offers support for you to acquire

- Problem-solving capabilities
- Computing concepts
- Skills with Java
- What it takes to make better software!
Java

• We will use the Java programming language to present many computer-science concepts
• You will get hands-on experience developing and debugging problem solutions using Java
• Java is an object-oriented language
• Java programs are compiled into machine language and tested

Pretest

• This pre-quiz will assess some capabilities presented in some sections of Intro to IT and College Algebra (see outcomes numbered 0.1, 1.0, 2.0, etc.)
• There will be more chances on all outcomes assessed
2. Course topics

1. Problem solving and system design
2. Computer organization and hardware
3. Introduction to Java
4. Standard Java data types
5. Branch statements and Boolean values
6. Writing and debugging loops (with arrays)
7. Methods, classes, and software development

CS I course-wide objectives

0a. Participate in class activities throughout the semester
0b. Solve problems as part of a team
0c. Present results in the classroom
0d. Present written results
0e. Show knowledge of facts and concepts
0f. Summarize the semester’s learning
0g. Write a documented programming project
0h. Reflect on reading of text material
Topic 1-4 objectives

1. Describe and apply principles of system specification and design
2. Explain and use the binary system of numerals and simple machine instructions
3. Write, document, and test a simple Java program
4. Use standard Java data types in documented, tested programs

Topic 5-7 objectives

5. Use Java branch statements and relational and logical operators.
6. Use and debug a variety of Java loops.
7. Use and explain Java features that support procedural and data abstraction.
3. How this course will deliver

- Have you taken courses organized around objectives or outcomes?
- Are you concerned about how learning is assessed in CS I?
- What are your expectations in course organization?

This course is an inquiry

- *FSU motto*: “Live to the truth”
- What does this mean?
- What is the truth?
- What does *truth* mean in software development and computer science?
- *Possible ways to the truth*:
  - Authority of experts
  - Our own experience, including experiments
  - Discussion
Classroom format

- Emphasis is on *inquiry, activity,* and *collaboration*
- *Slides and short presentations* summarize the content of the course
- We seek to create a *natural critical learning environment*
- *Your contribution and participation* matter

In this classroom

- We work on different problems
- We scratch our heads
- Brilliance means getting it after effort
- Everyone is brilliant sometimes
- Mistakes are intriguing
Course organization

• This course has seven *topics*
• Each has an *objective* and 3-5 *subtopics*
• Each subtopic has one or more *outcomes*
• Outcomes are considered *essential; priority; or challenge*
• A set of *problems* for each subtopic outcome is available
• Exercises and quizzes focus on these

Exercises, group work, and presentations

• *Exercises* consist of individual and group problem solving
• Each student solves problems from as many essential and priority outcomes as possible; I check off work received
• Each student prepares to lead a group in finding one solution per topic
• Exercises give you practice with quiz questions
Classroom exercise today

- Install the Java Development Kit (JDK) software environment (see handout)
- Edit Hello.java to add “+” and your name after the word “Hello”
- Compile and test
- Submit code and screen shot and test results
- For all exercises, please help others or ask for their help as needed

Group work in classroom

- Form groups of 3 students to take up a problem
- Each group should have
  - Facilitator – keeps discussion on track
  - Recorder – writes results of discussion
  - Reporter – presents results to class
- Participation by all in group work is one of our basic objectives in this course
Challenge: expression evaluator

- I would like teams to develop an expression evaluator for propositional-logic formulas
- Input: formula plus a set of variable assignments
- Example: \((p \land (q \lor \neg r), (t, f, t))\)
- Output: true or false
- I would like you to use this tool to investigate the satisfiability problem to see if any solutions, or solutions to any subsets of this problem, run efficiently

Assessing objectives in class

- After doing exercises on a topic, a student may show attainment of an outcome/objective by solving a quiz problem, in writing, in class
- More opportunities will be available for each outcome
- The main factor in success is attaining objectives and outcomes
- Another factor: contribution to everyone’s learning
Scoring answers, not quizzes

- A wrong answer means “try again”
- Scores on outcomes can only improve as you learn more
- Leaving a question blank or “I don’t know” often shows discernment
- Worst possible result is “Not yet”

What is a successful solution?

- Reflects understanding of
  - question or problem
  - relevant concepts
  - relevant procedures
- Usually reflects study and work on an exercise
Pretest

- This pre-quiz will assess some knowledge from Intro to IT and College Algebra
- Each pre-test question relates to one or more topics of this course
- Many of these outcomes are considered essential for CS I
- There will be multiple chances on all outcomes assessed

Final exam day and summary quiz

- During the last week of classes, we’ll have a *summary quiz* of multiple-choice questions and multi-topic problems
- On final exam day, students will present elements of their semester projects
Programming project

- As a semester project, you will construct a file-maintenance application in Java
- It will input, display, store, and retrieve data about something of your choice
- The project makes use of file I/O, class design, loops, methods, and arrays.
- The project goal is experience in specifications, design, coding, and testing

Assessment and grading

- To measure:
  - Individual achievement of learning objectives
  - Contribution to the learning of the class
- Breakdown: 60/40
- Assumptions: Learning is shared and measurable
Assessment of outcomes

Assumptions:
• Application of concepts is measurable via core and other topic objectives
• Facts about concepts matter
• We learn by summarizing and reflecting

Assessment of contribution and participation

We assume that learning happens by:
• Inquiring and sharing inquiry
• Being present
• Solving problems together
• Activity throughout the semester
Making this course easy or fun

- This course may or may not be that
- If you can’t find anything in it that matters to you, it may be hard and not fun
- If you can find something that grabs you, then you could be in luck
- It depends partly on finding that
- My job is to help you do so

Hidden curiosity and talent

- You were born curious; it’s in your nature
- Schooling can suppress curiosity
- One option is to allow our curiosity to re-emerge as part of our true selves
- For me, this can enable effortless effort and helps me to be present with what I study
The growth mindset

- Research results state:
  - People can learn new skills when they believe that their effort matters
  - Learning takes effort
  - Intelligence can grow with effort
- Alternative mindset: fixed
- The fixed mindset says that innate talent, not effort, is decisive and changeless

Quality and learning

- People enjoy doing quality work
- This requires freedom of choice and control of work environment
- Coercion and boredom may discourage quality work
- In school, part of learning is recognition of quality work
- Please evaluate your work and get your friends’ evaluations!
Who can judge quality?

- *Everyone* likes quality
- *You* can be a good judge if you know what to look for
- *Other students* may help you to see quality
- *Professor* may help you to see quality
- Research predicts that if we know quality, we will produce it

Grades, learning, and effort

- Learning requires *curiosity*, *intention to learn*, and *undistracted effort*
- Attention to grades distracts from what we’re learning
- If grades measure learning, then:
  - *Getting higher grades requires paying less attention to grades!*
Which matters: grades or content?

- Grades are opinions
- Opinions are fallible
- Course content could change our lives
- **Questions:**
  - Whose opinion defines what’s true?
  - When is it right to challenge a student?
  - When is it right to challenge a teacher?

Role of students in course development

- I organize the materials to make sense to me
- I revise topics as I present them
- Students often help to
  - clarify
  - correct
  - expand
  the course materials
**Academic integrity**

- Directly lifted text must be quoted and credited
- Use of ideas or other information must be credited by citations or references
- Citation standards for MLA and APA are given at [www.citationmachine.net](http://www.citationmachine.net)
- *Plagiarism*: “occurs when you use someone else’s ideas or words and represent them as your own.”
- See catalog for FSU policy

**A proposed agreement**

_I commit to:_
- know the course material, present it clearly
- return submitted work within a week.
- answer questions helpfully

_Students commit to:_
- ask questions
- answer reasonable questions, risking error
- submit work on time, even if incomplete
- work sometimes in groups
- present results or lead discussions
References


Peter J. Denning. Great principles of computing.  
*CACM* 46(11), November 2003, pp. 15-20.

