CSC 102-002, COMPUTER SCIENCE PRINCIPLES  Fall 2017
10 – 10:50, MWF Room 322 (McKibben Education Building)
Dr. David A. Cook, cookda@sfasu.edu, 303F McKibben, 468-2508

PREREQUISITE: Eligibility to enroll in College Algebra.

OFFICE HOURS: M/W 11-12:30, and 2:15 – 3:15. Tuesdays from 1-3:30, and Thursdays from 12:15 – 12:45. I am also available by appointment as needed. You should make an appointment to see me outside of office hours, but if my office door is open, you are free to drop in.. I do require appointments for Fridays.

COURSE DESCRIPTION
Fundamental concepts of computer systems, systems software, and an overview of computer science issues. Problem solving and program development using a high-level programming language.

COURSE PURPOSE
To introduce students to the basic concepts of computer systems, to fundamental systems software, to a disciplined approach to problem solving, to procedural program development in a high-level language, to software engineering principles, to ethics in computing, and to computer science careers.

COURSE CONTENT: TOPIC/(HOURS)
Introduction to computers (1)
Basic Concepts of Computer (3)
  Architectural overview
  Data storage and representation
  Computing environments
  Computer languages
Systems Software (6)
  Operating systems, editors, compilers
  Program linking, loading, and execution
  The Internet and electronic mail
Problem Solving Concepts (9)
  Strategies for problem solving
  Algorithm representation
Program Development (18)
  Use of a high-level programming language to solve simple problems on a computer
  Programming language concepts and features: primitive data types, expressions and operations, functions and parameters, fundamental control structures, one-dimensional arrays, text files
Software Engineering Principles (3)
  Life Cycle and Development Process
  Modular design and communication
  Documentation
Ethics and Careers (2)
Exams (3)
TOTAL  45

REQUIRED TEXTS: Introduction to Java, 10th Edition, by Y. Daniel Liang. The book is required reading - several test topics will not be covered in class, but are clearly explained in the book. You must have a copy (either hard copy or electronic) and you should read the book to reinforce class lectures. If a topic is covered in the book, and you come up to the office with a question, my first question back to you will be “Have you read the book?” I am happy to help you with questions – but you must make the effort to read the book. There are two versions of this book – a shorter custom version (used only in CSC 102) and a full version (for CSC majors). Both are available from the bookstore. I will discuss the differences on the first day of class.

EXAMINATIONS: Approximately 60% of the course grade – all exams are comprehensive, and include programming. I will give you a minimum of two weeks notice on the dates of each test.
Test 1 approximately 12% of your final grade, usually in late September.
Test 2 approximately 13% of your final grade, usually in mid to late October.
Test 3 approximately 15% of your final grade, usually in mid to late November.
Final Examination – Comprehensive, approximately 20% of your final grade
Note that the final is **Wed, 13 Dec 10:30 – 12:30. You must take the final to complete the course**

PROGRAMS: (Approximately 20% of the course grade)
There will be approximately 10 programming assignments of unequal weight. **You must complete (get a grade of 50% or greater) on half of the programs to pass this class. You must receive 50% or higher on all of the programs for an “A”. ALL PROGRAMS MUST BE TURNED IN ON D2L. NO EXCEPTIONS.** Programs are due at 10 PM, and I usually give you an extra 2-hour window. At midnight, the dropbox will close for late programs. This “late dropbox” gives you an extra 24 hours to submit a late program. Late programs must be working (no compile errors) to receive any credit – and late programs will only receive a maximum of 50%. Allowing students to submit programs outside of D2L (email, a USB drive) causes grading problems. I have a grader – and my grader can only see programs on D2L. **Again, I repeat – there are NO exceptions.** Submitting .~java or .class files count as 0. You do not have to have a computer to complete this class (although it is encouraged) – there are computers available on campus. Either a Mac or PC is acceptable. Chromebooks will not work.

HOMEWORK AND QUIZZES: (Approximately 10% of the course grade)
There might be a several homework assignments of unequal weight. Quizzes of unequal weight will be given during class. You may not make up a missed homework or a missed quiz.

ATTENDANCE: Attendance and constructive class participation are expected. I expect you to ask questions and participate in the class discussions. Class participation counts for 10% of your grade. If you continually don’t show up for class, you probably won’t get above a “B”.

EDUCATIONAL OBJECTIVES AND LEARNING OUTCOMES:

1. Demonstrate a fundamental knowledge of computer organization, computer operation, and the information hierarchy (binary numbers and character representations).
2. Apply the software life cycle to specific problems in such disciplines as business, mathematics, science, and engineering.
3. Perform problem analysis and program design using tools such as pseudocode, structure charts, and flowcharts.
4. Apply the features of a modern widely-used programming language in implementing solutions to well described problems. These features include declaration of data types and fundamental data structures, application of control structures (sequence, selection, repetition), utilization of I/O and file handling, development of structured program organization (subprograms with parameters), and inclusion of documentation.
5. Use operating systems tools (command system, editor, compiler, linker and loader) in single-user and/or multi-user environments.
6. Create appropriate test data and apply debugging and testing strategies.
7. Use E-mail, networks, and the Internet.
8. Demonstrate knowledge of fundamental computing terminology.
9. Demonstrate an understanding of the role of computing in society.

GRADING POLICY: My typical grading policy is the standard 90 cutoff for an A, 80 for a B, etc. Grades below 60 will receive an F. I will not raise the cutoffs. I reserve the right to lower the cutoffs by curving the grades – but the 90/80/70/60 cutoffs will never be raised. I will round all grades to the closest whole number.

REFERENCES: Horstmann, C., Big Java Late Objects, Wiley, 2013.

Syllabus Addendum, Fall 2017
Electronic Devices:
I encourage the use of electronic devices such as tablets, computers, etc. to facilitate your learning. Most of my slides will be posted on D2L prior to class – you are encouraged to download and bring them with you. Note that computers, tablets, phones, etc. are to be used to support learning in my class – not for social media updating, web browsing, texting, doing homework for other classes, etc. If it becomes obvious that you are not using your electronic devices properly and disrupting the learning of other students, I will ask you to stop. After the second warning, I will ask you to leave the class. If this becomes a continual problem, refer to my “ATTENDANCE” policy above – you will lose a letter grade. Note that all devices must be powered off, placed in a backpack or purse, and may not be used during tests. I ask that all phones be put on vibrate during class.
Hints to help you pass the class
(not an official part of the syllabus)

CSC 102 is a programming class – not a theory class. It requires you to demonstrate (via 8 programs) that you are able to program in Java. In addition, each test will also require you to write a program. If you do not need to take CSC 102 – please come and talk to me about taking CSC 101 or CSC 121 – which are NOT programming classes.

Many students underestimate the amount of time it takes to write a program. My assignments are always due at 10 PM, usually on a Friday. Waiting until 6 PM to start the program is not the way to pass this course. Most students who wait until the last minute inevitably have a compile error that they cannot quickly resolve – and while I try to reply to email in a timely manner, the evenings a program is due tend to be “heavy email” times. I cannot easily debug your program via email. Writing code is NOT like writing a report – it’s hard to estimate the time to complete, and it’s impossible to say, “This will only take me 4 hours”. Students learn this every semester. I try not to laugh. Late assignments are worth – at most – 50%. That’s only if they are perfect and within 24 hours. Coding is a skill – practice makes it easier and easier. If you don’t write code – you’ll have trouble on the tests.

In addition – you need to have access to a book. The book has many examples – and also a more complete coverage of topics I discuss during class. If you come in to my office asking for help – my first question will be “have you read the book?” If you reply that you don’t have a book – my ability to help you is limited. I’ll do my best to help – but the book is to be your first choice for help.

My office is not a place to write code – I will help you debug, help you find errors, walk you through the logic, and give you all the advice I can. However, every semester several students want to come in and write their program with me watching. Once we fix or work out your specific problem, I’ll ask you to go work on it elsewhere. If you have a later problem – come on back to see me again. This lets me assist other students in a timely manner. If you need to come back and see me 10 times in a day – that’s OK! Come to find out, I’m a nice person, and will not humiliate you. I make errors when I program – this lets me assist other students in a timely manner.

As mentioned in the syllabus – you need to complete half of the programs to pass, and turn in all to be eligible for an “A”. My grader clearly leaves your program grades (and comments) on D2L – you can always check your status. If you suddenly realize during the last week of the semester that you don’t have half of the programs completed – there is not much I can do.

You need to come to class – because I hit the “high points” and explain the complex topics during class. I will write many programs during class, using the programs to explain topics. While I usually either pass out (or put on D2L) copies of the programs – seeing me write the code and fix errors will help you understand how to fix your own errors. While attendance is not required, I DO take attendance every day. If you are having problems with a topic – and my attendance roster shows that you missed class that day – be aware that I do not reteach a lecture unless you have a valid medical excuse.

I cannot stress this enough – you need to come to class if you want a good grade. I try to make the class interesting and enjoyable. I also try to help any student who needs help. I cover material in class that you cannot easily get by reading the book. If you are having problems – please do not wait to come talk to me. Every topic in the class builds on previous topics. If you are having problems – the longer you wait, the further behind you will fall. Some of you have had this class before – and will feel the urge to skip because “You’ve seen it before.” This typically leads to “Well, I got the same grade I did before”. If you skip class and read the book, you might be able to get a “B”, at best. Skip class and don’t bother to read the book? You might be able to get a “D”, but not likely.

Missing a test is guaranteed to harm your grade. Each test is comprehensive, and counts anywhere from 12% to 20% of your grade. Missing one will drop your grade a minimum of a letter. Missing the final guarantees an “F”. You must have a valid excuse (cleared through Student Services) before I can let you retake a test. Let me epeat: Missed tests require some type of documented excuse.

It’s easier to write the programs that to copy and try to hide that you copied. Seriously – my grader and I can spot a copied program in about 10 seconds. I read EVERY program. And I’ve got 45 years’ experience helping me.

Want to do well in the class? Come to class, pay attention, do the programs, and take the tests. That’s it! Pretty simple, really, isn’t it?