1. Format
   1. Wednesday, September 28, 2016
      1. 50 min. for test
      2. Possibly an additional 10 min. or so at the end
   2. Need a pink scantron
   3. No devices or collaboration
   4. Test paper
   5. 100 questions
   6. Understanding of concepts
   7. 3 parts
      1. 25 T/F questions (2 pt. each)
      2. 6 multiple-choice questions (5 pts. each)
         1. Given code
            1. Purposefully not indented
            2. Must know what is what
         2. Questions about code
            1. What is operative?
            2. Possible bugs?

Compilation error? Runtime error? Execution error?

Program does not print anything out

* + 1. Write a simple program (20 points)
       1. Should take 5-10 min
       2. Neatly
  1. 3 tests
     1. 100 points each copy
     2. Each worth 25%
     3. One dropped
  2. Allowed to bring one 8.5” × 11” paper

1. Slides
   1. Slide Set #1: Basics
      1. Slide 18 🡪 end
         1. Components of a computer
         2. Basic outline and cycle of a C program
            1. Slides 23, 26
      2. No questions about how Code::Blocks works
      3. Basic Program Structure
         1. Comments
         2. **C Preprocessor Directives**
         3. main()
         4. Header files
         5. **Compilation vs. Execution vs. Logic errors**
   2. Slide Set #2: Variables, Operators
      1. All slides
      2. **Variables**: index to memory that OS takes care of.
         1. Types
            1. Characters
            2. Floating point values
            3. Double floating point values
            4. Integers

Signed

Unsigned

* + - 1. Declaration
      2. Initialization
      3. Legal names
      4. Variable identifier rules
         1. Naming rules
         2. Capitalization
      5. **Scope**: function in which a variable is defined.
         1. Local
      6. ASCII table
    1. Computer memory
    2. Operators
    3. 37 Keywords (concept)
    4. Basic binary arithmetic
       1. Bits
       2. Bytes
       3. Words
       4. Binary 🡪 Decimal
    5. **Constants**
    6. Operators
       1. **Simple operators**
          1. Know what they are
          2. How to use them
       2. Floating point division vs. integer division
          1. Integers: 2 types

/

% (mod)

* + - * 1. Floating point: only division giving decimals
      1. Precedence rule
      2. Boolean operators
    1. I/O functions
       1. printf()
       2. scanf()
          1. &

Necessary when reading char, int, float, double

**Not put when reading a string**

* + 1. Basic strings
       1. **String literals**: “”
       2. **Character literals**: ‘’
       3. char str [size of string]
  1. Slide Set #3: Conditionals and Loops (**most important**)
     1. 7 structures
        1. if
        2. if-else
        3. if-else-if-else
        4. switch
        5. for
        6. while
        7. do while
     2. Selection structures
     3. Repetition structures
     4. Role of {}
        1. 1 line: no need for {}
        2. >1 line: must use {}
        3. switch: no need for {}
     5. break
     6. ~~Random number functions~~ (Responsible for, but not on test)
        1. ~~srand()~~
        2. ~~rand()~~
  2. Slide Set #4: Programmer-defined functions
     1. Defining functions
        1. Prototype
        2. Header vs. Body
     2. Calling functions
     3. Passing functions
     4. Passing arguments to function
        1. Pass by value vs. pass by reference
     5. Returning values by functions
     6. Call by value vs. call by reference
     7. Programming styles