CSC-120/124
Review Outline

1. C Syntax Characters:
   - a. Statement terminator ;
   - b. Block characters {}
   - c. Parameter list characters ()
   - d. Array index characters []
   - e. Parameter separator ,
   - f. line comment // 2.2 pg 45
   - g. block comment /* */ 2.2 pg 45

   keywords 2.3 pg 46
   string constant " " 2.6 pg 49
   character constant ' '
   Declaration and initialization char c = 'A'; 1.4 pg 13
   int i = 1;

2. C Pre-Processor Directives:
   - a. Header file specification #include<> 1.5 pg 13 , 2.12 pg 59
   - b. Constant specification #define pi 3.14159 1.5 pg 13

3. C Data Types:
   - a. Integers:
     - i. char 1 byte
     - ii. short 2 bytes
     - iii. int 2 bytes / 4 bytes
     - iv. long 4 bytes
   - b. Real:
     - i. float 4 bytes
     - ii. double 8 bytes

4. C Operators: 2.8 pg 52
   - a. arithmetic:
     - i. multiplication *
     - ii. division /
     - iii. modulus %
     - iv. addition +
     - v. subtraction -
   - b. auto: 2.9 pg 53
     - i. increment ++ (pre/post)
     - ii. decrement - - (pre/post)

c. assignment: 2.10 pg 56
   i. +=
   ii. -=
   iii. *=
   iv. /=
   v. %=
   vi. ...
d. relational: <, >, <=, >=, == ,!= 3.1 pg 77
e. logical: !, &&, || 3.12 pg 97

5. C Statements:
   a. if 3.7 pg 85
   b. else
   c. for 3.10 pg 93
   d. while 3.8 pg 89
   e. do 3.14 pg 99
   f. switch 3.17 pg 103
   g. case
   h. break 3.16 pg 102
   i. continue
   j. return

6. C Arrays:
   a. one dimension x[row]
   b. two dimension x[col][row]

srand(time(NULL)); seed
rand() 0 -RAND_MAX (32767)
rand() % 50 0 -49
(rand() % 50) + 1 1 -50

7. C Addressing:
   a. value: x
   b. address: &x

8. C Place holders and control codes: 1.6
   a. place holder %
      i. string s
      ii. character c
      iii. integer d
      iv. real f
   b. code \
      i. \n newline
      ii. \r return
      iii. \t tab
      iv. \0 null
   c. Character 'character'
   d. String "string"

9. C Functions:
   a. prototype:
      return_value_type name (parameter_data_type_list);
   b. function:
      return_value_type name (parameter_data_type_and_name_list)
      {
         statements;
         return value;
      }
   c. input:
      i. scanf returns formatted data from stdin 1.6 pg 17
         scanf("format string", parameter list); stdio.h
      c, d, f, lf, Lf, s 1.6 pg 18

      ii. gets returns a line of text from the stdin
         gets(string); stdio.h
      iii. fflush clears the input buffer
         fflush(stdin); stdlib.h

   d. output:
      i. printf prints a formatted data to stdout 1.6 pg 15
         printf("format string", parameter list); stdio.h

      c, d, e, f, g, s 1.6 pg 16