1. (10 points) State whether each of the following statements is true or false. If false, explain why.

   a) Declarations can appear anywhere in the body of a function.
   False. Declarations must be placed before the left brace of a function and before any executable statements.

   b) The equality operators (e.g. ==) have higher precedence level than the relational operators (e.g. <).
   False. The equality operators (e.g. ==) have lower precedence level than the relational operators.

   c) Every C statement ends with a period.
   False. Every C statement ends with a semicolon.

   d) A switch statement will not compile without a default case.
   False. Default case in a switch statement is optional. It is used to catch exceptional cases.

   e) The expression (x > 1 || y < 0) is true if x > 1 is true and y < 0 is true.
   False. ’||’ means logical OR. The expression (x > 1 || y < 0) is true if one of the following cases are true:
   1) x > 1 is true
   2) y < 0 is true
   3) Both x > 1 and y < 0 are true.

2. (10 points) Fill in the blanks in each of the following:

   a) All programs can be written in terms of the three control structures: sequence, selection, and repetition.

   b) When executed in a repetition structure, the continue statement, causes the next iteration of the loop to be performed immediately, and the break statement causes an immediate exit from the current loop.

   c) A special value used to indicate the “end of data entry” is called a sentinel or a flag value.
d) Variable names in C may not begin with a digit or a punctuation (other than underscore).

e) Use a while or do-while loop for event-controlled repetition.

3. (15 points) Evaluate the following expressions and show all your work:

a) (2 points) Given an integer \( b = 3 \), what is the numeric value of “\(!b\)”?

Solutions:

\[ !b = !3 = 0 \]

b) (2 points) Given integers \( b, c, \) and \( d \), where \( c = 2, d = 3 \). What is the value of \( b \) after the following assignment?

\[ b = d/c ; \]

Solutions:

\[ b = 3/2 \]
\[ = 1 \]

(Because this integer division, the fraction part is truncated.)

c) (2 points) Given integers \( a, b, c, \) and \( d \), where \( a = 2, b = 3, c = 4 \). What is the value of \( d \) after the following assignment?

\[ d = 1 + a * b % c ; \]

Solutions:

\[ d = 1 + 2 * 3 % 4 \]
\[ = 1 + 6 % 4 \]
\[ = 1 + 2 \]
\[ = 3 \]

d) (4 points) Given integers \( b, c, \) and \( d \), where \( b = 3, c = 4, d = 5 \). What are the values of all these variables after executing the following statement?

\[ b *= c = d + 2 ; \]

Solutions:

Assignment operators: evaluate from right to left.

\[ b *= c = 5 + 2 \]
\[ b *= c = 7 \]

\[ b *= 7 \]
\[ b = b* 7 \]
\[ = 3 * 7 \]
\[ = 21 \]

\[ d = 5 \]
e) (5 points) Given integers \( a, b, c, d \) and \( e \), where \( a = 2, b = 3, c = 4, d = 5 \). What are the values of all these variables after the following statement?

\[
e = \text{--}b / c + a * d++ ;
\]

**Solutions:**
Unary operators have higher precedence level than others. Decrement \( b \) before evaluation and increment \( d \) after the evaluation is completed.

\[
e = \text{--}3 / 4 + 2 * 5++ \quad \rightarrow \quad a = 2, \quad c = 4
\]
\[
= 2 / 4 + 2 * 5 \quad \rightarrow \quad b = 2, \quad d = 6
\]
\[
= 0 + 2 * 5
\]
\[
= 0 + 10
\]
\[
= 10 \quad \rightarrow \quad e = 10
\]

4. (20 points) Write C statements that do the following:

a) Print the following text:

\[
\% : \text{for modulus calculation.}
\]

\[
\text{printf("}\%\% : \text{for modulus calculation.}\n") ;}
\]

b) Given \( a \) is a floating variable, \( b \) and \( c \) are integer variables. Input \( a, b, \) and \( c \) from the keyboard using scanf.

\[
\text{scanf("}\%f\%d\%d", \ a, \ b, \ c) ;}
\]

c) Write a ‘for’ statement that prints the sequence of values: “30 28 26 24 22 20”.

\[
\text{for (}i = 30 ; \ i >= 20 ; \ i -= 2\} \{
\text{printf("}\%d \", \ i) ;}
\]

d) Write a ‘switch’ statement to test whether an integer value, \( x \), is an even or odd number. If \( x \) is 3, the printed text should look like “\( x = 3 \) is an odd integer”. If \( x \) is 4, the printed text should look like “\( x = 4 \) is an even integer”.

\[
\text{switch} \ (x \% 2) \{
\text{case} \ 0 :
\text{printf("}x = \%d \text{ is an even integer}\n", \ x) ;
\text{case} \ 1 :
\text{printf("}x = \%d \text{ is an odd integer}\n", \ x) ;}
\}
5. (15 points) What is the output of the following code:

a) Assume that the next input line contains the “1 2 3”.

```c
int a = 4;
int b;
int c;
printf("Enter three integers:\n");
scanf("%d,%d,%d", &a, &b, &c);
printf("a = %d
b = %d
c = %d
", a, b, c);
```

**Output:**
Enter three integers:
a = 1
b = "garbage"
c = "garbage"

b) for ( i = 1; i < 3; i++ ) {
    for ( j = 1; j < 4; j++ ) {
        if ( j % 2 == 0 ) {
            printf("X") ;
        } else {
            printf("O") ;
        }
    }
    printf("\n") ;
}
printf("Done.\n") ;

**Output:**
OXO
OXO
OXO
Done.
c) for ( i = 1; i < 6; i++ ) {
    if (i == 3) {
        continue;
    }
    printf("%d\n", i);
}
printf("Done.\n");

Output:
1
2
4
5
Done.

6. (30 points) Identify and correct errors in each of the following C code fragments:

a) /* printf("x + y = %d, x + y ) */

   Error:
   1. Starting a comment line with */.

   Correction:
   /* printf("x + y = %d, x + y ) */

b) if (number => largest );
    largest == number ;

   Error:
   1. The syntax for the relational operator, =>, is incorrect.
   2. The syntax for the assignment operator is incorrect.

   Correction:
   if (number >= largest );
       largest = number ;
c) counter = 2;
  Do {
    if ( counter % 2 == 0 )
      printf( "%.d\n", counter);
    counter += 2;
  } While (counter < 100) ;

Error:
1. Keyword ‘Do’ should not be capitalized.
2. ‘if’ statement is redundant.
3. Extra ‘.’ in the format control specifier, ‘%d’.
4. Keyword ‘While’ should not be capitalized.

Correction:
counter = 2;
do {
  printf( "%d\n", counter);
  counter += 2;
} while (counter < 100) ;


d) while ( y > 0 ) {
  printf ("%d\n", y);
  ++y;
}

Error:
1. Keyword ‘While’ should not be capitalized.
2. This while loop is an infinite loop.
   Need to either decrement y or
   change the condition for the while loop.

Correction:
while ( y > 0 ) {
  printf ("%d\n", y);
  --y;
}
e) Assuming that the next input line contains “100 A”, the code fragment below should print the following text:

```
Integer is 100
Character is A
```

```c
    scanf( "%d", &intVal );
    charVal = getchar();
    printf( "Integer is %d\nCharacter is %c\n", intVal, charVal );
```

**Error:**

1. Standard library function, 'Scanf', should not be capitalized.
2. Need to insert extra read statement for reading a space or 'A'.
3. The colon, ':', inside of the printf statement should be changed to ' is'.

**Correction:**

```
    scanf( "%d", &intVal );
    charVal = getchar();
    charVal = getchar();
    printf( "Integer is %d\nCharacter is %c\n", intVal, charVal );
```
for (f = 0.00001; f <= 0.0001; f += 0.00001) 
printf("%.6f\n", f);

Error:
1. Use floating number to control a for loop.
   Due to round off errors (inaccuracy of floating point),
   the number of executions of the loop may depend on the floating
   point hardware. It would be better to use an integer loop and
   perform the proper calculation in order to get the proper results.

Correction:
for (i = 1; i <= 10; i++)
printf("%.6f\n", i*0.00001);