Covering:

Chapter 1 to Chapter 5

1. What is wrong with the following code?
```
#include <stdio.h> /*This is a directive line*/
main /* This function will be automatically called*/
(void/* I do not have arguments to pass*/)
{ /* This is the first line in the program
  /* I am so excited to write this program */
  printf("Hello world\n")}
  /* Finally I am done with it
  /* But I am not sure why it does not work :-( */
}
```

2. Which of the following are legal C identifier?
```
100_bottles, _100_bootles, one-hundred-bottles, one_hundred_bottles,
one_hundred_bottles_, one__hundred__bottles, one__hundred__bottles__, If, main,
Printf, and while.
```

3. How many token are there in the following C statement?
```
x=answer=(a*b/++c+6-a%f)/(k);
```

4. What is the output of the following C statement?
```
printf("%6d, %4d\n" , 86, 1040);
```

5. What is the output of the following C statement?
```
printf("%12.5e\n", 30.253) ;
```

6. What is the output of the following C statement?
```
printf("%.4f\n", 83.162) ;
```

7. What is the output of the following C statement?
```
printf("%-6.2g\n", .0000009979);
```

8. Write calls to `printf` that display a `float` variable x in the following formats:

   (a) Exponential notation; left-justified in a field of size 8; one digit after the decimal point.

   (b) Exponential notation; right-justified in a field of size 10; six digits after the decimal point.

   (c) Fixed decimal notation; left-justified in a field of size 8; three digits after the decimal point.

   (d) Fixed decimal notation; right-justified in a field of size 6; no digits after the decimal point.

9. What is the output of the following C statement?
```
printf("*%d*\n*%-d*\n*%d*\n*%-d*\n",123456,123456,-123456,-123456);
```

10. What is the output of the following C statement?
```
printf("*%.10d*\n*%-.10d*\n*%.10d*\n*%-.10d*\n",
       123456,123456,-123456,-123456);
```

11. What is the output of the following C statement?
```
printf("*%20d*\n*%-20d*\n*%20d*\n*%-20d*\n",
       123456,123456,-123456,-123456);
```

12. What is the output of the following C statement?
```
printf("*%20.10d*\n*%-20.10d*\n*%20.10d*\n*%-20.10d*\n",
       123456,123456,-123456,-123456);
```

13. What is the output of the following C program?
```
#include <stdio.h>
main(void)
{ float m = 6, n = 6, o = 6, p = 6, q = 6;
  int i = 3, j = 3;
  m ++; ++ m;
  m += n += o -= p *= q /= i %= ++j ;
  printf("%f\n%f\n  %f\n  %f\n %f\n %d\n%d", m, n, o, p, q, i, j);
  m += 3 + (n += 3 + (o -= 3 + (p *= 3 + (q /= 3 + (i %= j++)))));
  printf("%f %f\n%f %f\n%f %d %d\n", ++m, n++, --o, p--, ++q, i--, --j);
}
```

14. Suppose that we call scanf as follow:

```
scanf ("%d%f%d" , &i, &x, &j);
```

If the user entered
```
10.3 5 6
```
what will be the values of i, x, and j after the call? (Assume that i and j are int variables and x is a float variable.)

15. Suppose that we call scanf as follow:

```
scanf("%f%d%f", &x, &i, &y);
```

If the user entered
```
12.3 45.6 789
```
what will be the values of x, i, and y after the call? (Assume that x and y are float variables and i is an int variable.)

16. For each of the following pairs of scanf format strings, indicate whether or not the two strings are equivalent. If they're not, show how they can be distinguished.
```
(a) "%d" versus " %d"
(b) "%d-%d-%d" versus "%d -%d -%d"
(c) "%f" versus "%f "
(d) "%f,%f"   versus "%f, %f"
```

17. Write a program that formats products information entered by the user. A program session should be as follow:
```
Enter item number: 583
Enter unit price: 13.5
Enter purchase date (mm/dd/yyyy): 09/24/2008

Item                    Unit                    Purchase
                        Price                   Date
583                     $ 13.50                 09/24/2008
```
The item number and date should be left justified; the unit price should be right justified.
Allow dollar amounts up to $9999.99. Hint: Use tabs (i.e., "\t") to line up the columns.

18. Write a program that prompts the user to enter a ten digit telephone number in the form xxxxxxxxxx and then displays the number in the form (xxx) xxx-xxxx.

19. Write a program that asks the user to enter the numbers from 1 to 16 (in any order) and then displays the numbers in the same order in a 4 by 4 row-wise arrangement, followed by the sums of each rows, each columns, and each diagonals.

20. Show the output produced by the following program fragment.
```
int i, j;
i = 7; j = 8;
i *= j + 1;
printf("%d %d\n", i, j);
```

21. Show the output produced by the following program fragment.
```
int i; int j, k;
i = j = k = 1;
i += j += k;
printf("%d %d %d\n", i, j, k);
```

22. Show the output produced by the following program fragment.
```
int i; int j; int k;
i = 1; j = 2; k = 3;
i -= j -= k;
printf("%d %d %d\n", i, j, k);
```

23. Show the output produced by the following program fragment.
```
int i, j, k;
i = 2; j = 1; k = 0;
i *= j *= k;
printf("%d %d %d\n", i, j, k);
```

24. Show the output produced by the following program fragment.
```
int i, j;
i = 6 ;
j = i += i;
printf("%d %d", i, j);
```

25. Show the output produced by the following program fragment.
```
int i, j;
i = 5 ;
j = (i -= 2) + 1;
printf("%d %d", i, j);
```

26. Show the output produced by the following program fragment.
```
int i, j;
i = 7 ;
j = 6 + (i = 2.5) ;
printf("%d %d", i, j);
```

27. Show the output produced by the following program fragment.
```
int i, j;
i = 2; j = 8;
j = (i = 6) + (j = 3);
printf("%d %d", i, j);
```

28. Show the output produced by the following program fragment.
```
int i;
i = 1 ;
printf("%d ", i++ - 1);
printf("%d", i);
```

29. Show the output produced by the following program fragment.
```
int i, j;
i = 10; j = 5;
printf("%d ", i++ - ++j);
printf("%d %d", i, j);
```

4 11 6

30. Show the output produced by the following program fragment.
```
int i, j;
i = 7 ; j = 8 ;
printf("%d ", i++ - --j);
printf("%d %d", i, j);
```

0 8 7

31. Show the output produced by the following program fragment.
```
int i, j, k;
i = 3; j = 4; k = 5;
printf("%d ", i++ - j++ + --k);
printf("%d %d %d", i, j, k);
```

3 4 5 4

32. Show the output produced by the following program fragment.
```
int i, j;
i = 5;
j = ++i * 3 - 2;
printf("%d %d", i, j);
```

6 16

33. Show the output produced by the following program fragment.
```
int i, j;
i = 5 ;
j = 3 - 2 * i++;
printf("%d %d", i, j);
```

6 -7

34. Show the output produced by the following program fragment.
```
int i, j;
i = 7 ;
j = 3 * i-- + 2;
print f("%d %d", i, j);
```

6 23

35. Show the output produced by the following program fragment.
```
int i, j;
i = 7 ;
j = 3 + --i * 2;
printf("%d %d", i, j);
```

6 15

36. Write a program that computes the volume of a sphere (4/3 π r3), where *r* is the radius of the sphere. Your program should ask the user to enter the radius of the sphere.

37. Write a program that asks the user to enter a value for *x* and then displays the value of the following polynomial:
$3x^5 + 2x^4 - 5x^3 + x^2 - x - 6$

38. Write a program that asks the user to enter a dollar amount and then shows how to pay that amount using the smallest number of $20, $10, $5, $2, and $1.

39. Write a program that asks the user to enter a two-digit number, then prints the number with its digits reversed. A session with the program should have the following appearance:
```
Enter a two-digit number: 28
The reversal is: 82
```
Hint: Read the number using `%d`, then break it into two digits. If `n` is an integer, then `n % 10` is the last digit in `n` and `n / 10` is `n` with the last digit removed.

40. Write a program to read a `float` representing a number of degrees Celsius, and print as a `float` the equivalent temperature in degrees Fahrenheit. Print your results in a form such as
```
Enter a temperature is Celsius degrees: 100
100.0 degrees Celsius is equivalent to 212.0 degrees Fahrenheit
```

41. Write a program that asks the user to enter an integer number of seconds, then prints as output the equivalent time in hours, minutes and seconds.

42. Where possible, write equivalents for the following statements using compound assignment operators:
```
s = s / 5;
q = q * n + 4;
z = z - x * y;
t = t + (u % v);
```

43. What values are assigned to `n`, `m`, and `p`, given these initial values?
```
i = 4; j = 8;
n = ++i * --j;
m = i + j--;
p = i + j;
```

44. What are the values of `n`, `m`, and `p` after the execution of this code fragment?
```
j = 5; k = 2;
n = j - ++k;
m = j-- + k--;
p = k + j;
```

45. What arc the values of `x`, `y`, and `z` after the execution of this code fragment?
```
x = 3; y = 5; z = 2;
x *= y+ z;
y /= 2 * z + 1;
z += x;
```

46. Show the output produced by the following program fragment.
```
int i = 1; int j = 2;
i -= - -- j;
printf("%d %d\n", i, j);
```

47. Show the output produced by the following program fragment.
```
int i = 1;
int j = 2;
i -= - - - j;
printf("%d %d\n", i, j);
```

48. Show the output produced by the following program fragment.
```
int i = 1;
int j = 2;
i = + + + + + j;
j = - - - - - j;
printf("%d %d\n", i, j);
```

49. Show the output produced by the following program fragment.
```
int i = 1;
int j = 2;
i = - + - + - + j;
printf("%d %d\n", i, j);
```

50. Show the output produced by the following program fragment.
```
int i = 1;
int j = 2;
i = + - + - + - j;
printf("%d %d\n", i, j);
```

51. Show the output produced by the following program fragment.
```
int i = 1;
int j = 2;
i = - + - + - ++ j;
printf("%d %d\n", i, j);
```

52. Show the output produced by the following program fragment.
```
int i = 1;
int j = 2;
i = + - + - + -- j;
printf("%d %d\n", i, j);
```

53. Show the output produced by the following program fragment.
```
int i = 1, j = 2;
int k = 3, m = 4;
i += j - (k *= m++);
printf("%d %d %d %d\n", i++, ++j, k--, --m);
```

54. Show the output produced by the following program fragment.
```
int i = 1, j = 2, k = 3, m = 4;
i *= --j * (k *= --m);
printf("%d %d %d %d\n", i++, ++j, k--, --m);
```

55. Show the output produced by the following program fragment.
```
int i = 1, j = 2;
int k = 3, m = 4;
i %= j++ % (k += --m);
printf("%d %d %d %d\n", i++, ++j, k--, --m);
```

56. Show the output produced by the following program fragment.
```
int i = 1, j = 2;
int k = 3, m = 4;
i *= j / - (k -= ++m);
printf("%d %d %d %d\n", i++, ++j, k--, --m);
```

57. If the value of n is 4 and m is 5, will the value of ++(n * m) be 21? Explain your answer.

58. In which order the following operators will be executed:
```
i < j || j < k || k < l
i < j && j < k && k < l
i < j || j < k && k < l
i < j <  j < k <  k < l
i < j >  j < k >  k < l
i < j == j < k && k < l
i < j != j < k || k < l
```

59. Show the output produced by the following program fragment.
```
int i = 2, j = 3, k = i * j == 6;
printf("%d\n", k);
```

60. Show the output produced by the following program fragment.
```
int i =  5, j = 10, k = 1;
printf("%d\n", k > i < j);
```

61. Show the output produced by the following program fragment.
```
int i = 3, j = 2, k = 1;
printf("%d\n", i  < j == j < k);
```

62. Show the output produced by the following program fragment.
```
int i = 3, j = 4, k = 5;
printf("%d\n", i % j + i < k);
```

63. Show the output produced by the following program fragment.
```
int i = 3, j = 4, k = 5;
printf("%d\n", i % j + i < k );
```

64. Show the output produced by the following program fragment.
```
int i = 3, j = 4, k = 5;
printf("%d\n", k = i * j == 6 );
```

65. Show the output produced by the following program fragment.
```
int i = 3, j = 4, k = 5;
printf("%d\n", i < j == k < i );
```

66. Show the output produced by the following program fragment.
```
int i = 3, j = 4, k = 5;
printf("%d\n", k > i < j );
```

67. Show the output produced by the following program fragment.
```
int i = 3, j = 4, k = 5;
printf("%d\n", i < j  || ++ j < k );
printf("%d %d %d\n", i, j, k );
```

68. Show the output produced by the following program fragment.
```
int i = 7, j = 8, k = 9;
printf("%d\n", i - 7 && j ++ < k );
printf("%d %d %d\n", i, j, k );
```

69. Show the output produced by the following program fragment.
```
int i = 7, j = 8, k = 9;
printf("%d\n", ( i = j ) || ( j = k ));
printf("%d %d %d\n", i, j, k );
```

70. Show the output produced by the following program fragment.
```
int i = 1, j = 1, k = 1;
printf("%d\n", ++i || ++ j && ++ k );
printf("%d %d %d\n", i, j, k );
```

71. Show the output produced by the following program fragment. Assume that i, j, and k are int variables and you compile the segment using C99. What will be the produced output if i, j, and k are bool?
```
i = 0; j = 1; k = 2;
printf("%d %d %d\n", i, j, k);
printf("%d %d %d\n", i--, j--, k--);
printf("%d %d %d\n", i, j, k);
```

72. Show the output produced by the following program fragment. Assume that i, j, and k are int variables and you compile the segment using C99. What will be the produced output if i, j, and k are bool?
```
i = 0; j = 1; k = 2;
printf("%d %d %d\n", i, j, k);
printf("%d %d %d\n", i++, j++, k++);
printf("%d %d %d\n", i, j, k);
```

73. Show the output produced by the following program fragment. Assume that i, j, and k are int variables and you compile the segment using C99. What will be the produced output if i, j, and k are bool?
```
i = 0; j = 1; k = 2;
printf("%d %d %d\n", i, j, k);
printf("%d %d %d\n", --i, --j, --k);
printf("%d %d %d\n", i, j, k);
```

74. Show the output produced by the following program fragment. Assume that i, j, and k are int variables and you compile the segment using C99. What will be the produced output if i, j, and k are bool?
```
i = 0; j = 1; k = 2;
printf("%d %d %d\n", i, j, k);
printf("%d %d %d\n", ++i, ++j, ++k);
printf("%d %d %d\n", i, j, k);
```

75. Is the following if statement legal?

```
if (n >= 1 <= 10)
    printf("n is between 1 and 10 \n");
```

If so, what does it do when n is equal to 5 and when n is equal 15?

76. What does the following statement print if i has the value of 17? What does it print if i has the value 31?

```
printf("%d\n", ++i >= 0 ? ++i : --i);
```

77. What does the following statement print if i has the value of 17? What does it print if i has the value 31?

```
printf("%d\n", ++i >= 0 ? ++i : (i-=10));
```

78. The following `if` statement is unnecessarily complicated. Simplify it as much as possible. (Hint: the entire statement can be replaced by a single assignment.)
```
if( age >= 13)
   if( age <= 19)
      teenager = true;
   else
      teenager = false;
else
   if( age < 13)
      teenager = false
```

79. Convert the following conditional expression to traditional if statement
```
i = j < k ? l > m ? 1 : 2 : n == p ? 3 : 4;
```

80. What value is assigned to `fee` by the `if` statement when `speed` is `75`?
```
if(speed > 35)
   fee = 20.0;
else if(speed > 50)
   fee = 40.00;
else if(speed > 75)
   fee = 60.00;
```

81. What value is assigned to `fee` by the `if` statement when `speed` is `75`?
```
if(speed > 75)
   fee = 60.0;
else if(speed > 50)
   fee = 40.00;
else if(speed > 35)
   fee = 20.00;
```

82. Which `if` statement in the above two questions seems reasonable?

83. What output does the following program fragment produce? (Assume that `i` is an integer variable)

```
i = 1;

switch (i % 3)
{
   case 0: printf("zero");
   case 1: printf("one");
   case 2: printf("two");
}
```

84. What will be printed by this `switch` statement if the value of color is `'R'`?
```
switch(color)
{
   case 'R':
         printf("red\n");
   case 'B':
         printf("blue\n");
   case 'G':
         printf("green\n");
}
```

85. What output line(s) are displayed by the statements that follow when `grade` is `'I'`? When `grade` is `'B'`? and When `grade` is `'b'`?

```c
switch(grade)
{
  case 'A':
        points = 4;
        break;
  case 'B':
        points = 3;
        break;
  case 'C':
        points = 2;
        break;
  case 'D':
        points = 1;
        break;

  case 'E':
  case 'I':
  case 'W':
        points = 0;
}
if(points > 0)
  printf("Passed, points earned = %d\n", points);
else
  printf("Failed, no points earned\n");
```

86. Explain the difference between the following two programs. For each group of statements, give the final value of `x` if the initial value of `x` is `1`.
```c
if(x >= 0)
  x = x + 1;
else if(x >= 1)
  x = x + 2;

if(x >= 0)
  x = x + 1;
if(x >= 1)
  x = x + 2;
```

87. Study the following program fragment,
```c
if(x==1)
if(y==2)
if(z==3)
printf("a\n");
else
printf("b\n");
else
printf("c\n");
printf("d\n");
```
what are the range of values of `x`, `y`, and `z`, that will lead to the execution of each `printf` statement?

88. Write a multiple-alternative `if` statement to display a message indicating the educational level of a student based on the student's number of years of schooling (0, none; 1-5, elementary school; 6-8. middle school; 9-12. high school; more than 12, college). Print a message to indicate bad data as well.

89. Write a `switch` statement to select an operation based on the value of `inventory`. Increment `total_paper` by `paper_order` if `inventory` is `'B'` or `'C'`; increment `total_ribbon` by `ribbon_order` if `inventory` is `'E'`, `'F'`, or `'D'`; increment `total_label` by `label_order` if `inventory` is `'A'` or `'x'`. Do nothing if `inventory` is `'M'`. Display an error message if the value of `inventory` is not one of these eight letters.

90. Write an `if` statement that displays an acceptance message for an astronaut candidate if the person's weight is between the values of `opt_min` and `opt_max` inclusive, the person's age is between `age_min` and `age_max` inclusive, and the person is a nonsmoker (`smoker` is `false`).

91. Write a `switch` statement that assigns to the variable `lumens` the expected brightness of a standard light bulb whose wattage has been stored in `watts`. Use this table:

Watts:	15	25	40	60	75	100
Brightness (in Lumen):	125	215	500	880	1000	1675

Assign `-1` to `lumens` if the value of `watts` is not in the table.

92. Write a nested `if` statement equivalent to the `switch` statement described in the previous exercise.