You may take this test with you afterwards, but you must turn in your answer sheet.

This test has the following sections:
   I. True/False .......................... 20 points; (10 questions, 2 points each)
   II. Multiple Choice ................ 80 points; (20 questions, 4 points each)

100 points total

This test is worth 10% of your final grade. You must put your answers on the bubble form. This test is open book and open notes, but no computers. For the multiple choice problems, select the best answer for each one and select the appropriate letter on your answer sheet. Be careful - more than one answer may seem to be correct. Some questions are tricky.

**True/False: (1 point each)**  On your bubble form fill out A for true and B for false.

1. The return type of main() is supposed to be int in C++.
   - T

2. While a C program would use a scanf statement to read input from the keyboard into a variable, a C++ program could use either the scanf statement or the cin statement.
   - T

3. To read in two values into two variables, two separate scanf statements are needed.
   - T

4. A single printf statement in a program always generates a single line of output.
   - T

5. A switch-case statement can always be rewritten with multiple if-else statements.
   - T

6. Multiple if-else statements can always be equivalently rewritten using a switch-case statement.
   - T

7. Generally an if-else statement can be rewritten using two if statements.
   - T

8. The following code automatically figures out the length of the array and allocates enough memory for it.
   ```c
   int numbers[] = {1,2,3,4,5,6,7};
   ```
   - T

9. The following code allocates space for the 3 characters and in addition automatically adds a space for a NULL character which is automatically added to the end of the array.
   ```c
   char word1[] = "ABC";
   ```
   - T

10. The following code will display the text: First letter is: D
    ```c
    char letters[] = "DFR";
    printf("First letter is: \%c", letters[ 1]);
    ```
    - F
11) Consider the exercise we did the first day of class, where we explored different methods of counting the number of students in the room. One approach we used was for a person to find another person standing, compare how many “stored” values each had, have one person sum them and represent that total, and then the other person sat down. This was then repeated. What was the advantage of this approach that we discussed?

   a) It is more accurate than the other approaches we explored
   b) Everyone gets to participate and actively learn
   c) The approach is similar to what computers do
   d) This method of counting illustrates a process being scalable

12) What is the relationship between programming in C and programming in C++?

   a) They are different names for the same thing
   b) A C compiler can run any C++ program
   c) A C++ compiler can run any C program
   d) They require different compilers

13) What is the point of the video shown in class of a man eating a bicycle?

   a) Even difficult problems can be solved using a computer
   b) If you are stuck on a problem, it can usually be solved from different angles.
   c) Technological problems require people to solve them
   d) Problems should be broken down with pieces solved separately

14) Consider the two programs shown below:

Option A:

```c
#include <stdio.h>
int main()
{
    printf("First One\n");
    return 0;
}
```

Option B:

```c
#include <iostream>
using namespace std;
int main()
{
    cout << "Second One\n";
    return 0;
}
```

Which of the following is the best answer regarding the above two programs A and B?

   a) Neither A nor B will compile and run.
   b) A will compile and run, but B will not
   c) B will compile and run, but A will not
   d) Both A and B will compile and run
15) Consider the two programs shown below:

Option A:

```c
#include <stdio.h>

int main()
{
    int value;
    printf("Enter value: \n");
    scanf("%d", value);
    printf("+1 is:%d", value+1);
    return 0;
}
```

Option B:

```c
#include <iostream>

using namespace std;

int main()
{
    int value;
    cout << "Enter value:
";
    cin >> value;
    cout << "+1 is:
" << value+1;
    return 0;
}
```

Which of the following is the best answer regarding the above two programs A and B?

a) Neither A nor B will compile and run as expected.
b) A will compile and run as expected, but B will not
c) B will compile and run as expected, but A will not
d) Both A and B will compile and run as expected.

16) Consider a program where an array is passed as a parameter to a function. Upon returning from the function call, assume you notice that another variable’s value has now changed, even though it is declared locally within main() and not passed to the function.

What is the best explanation for this?

a) It is human error. Your program does, in fact, have instructions that modify that variable.
b) Your input has an extra carriage return that is causing problems with your code.
c) You are overwriting the end of the array within the function.
d) None of the above

17) Consider the following statements about using functions in a program:

I. Functions make it easier to reuse code
II. Functions make programs run faster
III. Functions make it easier to break up a problem into pieces

Which of the above are true statements about functions?

a) II only
b) I and II
c) I and III
d) I, II and III
18) Consider the code segment shown below:

```c
i=0;
while( i<9 ) {
    printf("%d ", i);
    i++;
}
printf("\n");
```

Which of the following two code segments will give the same output as the above code?

Option A:
```c
for( i=0; i<9; i++ ) {
    printf("%d ", i);
}
printf("\n");
```

Option B:
```c
i=0;
do {
    printf("%d ", i);
    ++i;
} while( i<9);
printf("\n");
```

a) Neither A nor B will give the same output.
b) A will give the same output, but B will not
c) B will give the same output, but A will not
d) Both A and B will give the same output.

19) Consider the three cases below, and three types of loops: for, while, and do-while:
   I. Displaying a menu and getting the input, verifying it is a valid menu option
   II. Checking a logical condition to decide whether some code should be executed at all
   III. Repeating code a known number of times

Which type of loop is the best match for each of the above cases I, II and III?

a) I. do-while   II. while   III. for
b) I. while     II. while   III. for
c) I. while     II. do-while III. while
d) I. while     II. for      III. for
20) Consider the following statement:
   
   \[
   \text{salary} = (\text{age} > 21) ? 15 : 13;
   \]

   This is equivalent to which of the following sections of code?

   \begin{align*}
   \text{A) } & \quad \text{if (salary > 15)} \\
   & \quad \text{salary = age;}
   \text{else } \\
   & \quad \text{salary = 13;} \\
   \text{B) } & \quad \text{if (salary = (age > 21))} \\
   & \quad \text{salary = 15;}
   \text{else } \\
   & \quad \text{salary = 13;} \\
   \text{C) } & \quad \text{if (age > 21)} \\
   & \quad \text{salary = 15;}
   \text{else } \\
   & \quad \text{salary = 13;} \\
   \text{D) } & \quad \text{if (age > 21)} \\
   & \quad \text{salary = 13;}
   \text{else } \\
   & \quad \text{salary = 15;}
   \end{align*}

   For the following three problems, consider the following four alternatives of types of code, where the layout helps convey what the code is doing:

   \begin{align*}
   \text{A) } & \quad \text{if (expression1)} \\
   & \quad \text{action1;}
   \text{else if (expression2)} \\
   & \quad \text{action2;}
   \text{else if (expression3)} \\
   & \quad \text{action3;}
   \text{B) } & \quad \text{if (expression1)} \\
   & \quad \text{action1;}
   \text{if (expression2)} \\
   & \quad \text{action2;}
   \text{if (expression3)} \\
   & \quad \text{action3;}
   \text{C) } & \quad \text{switch (variable) } \\
   & \quad \{ \\
   & \quad \text{case 1: action1;}
   & \quad \text{break;}
   & \quad \text{case 2: action2;}
   & \quad \text{break;}
   & \quad \text{case 3: action3;}
   & \quad \text{break;}
   & \quad \}
   \text{D) } & \quad \text{if (expression1)} \\
   & \quad \text{action1;}
   \text{else } \\
   & \quad \text{if (expression2)} \\
   & \quad \text{action2;}
   \text{else } \\
   & \quad \text{if (expression3)} \\
   & \quad \text{action3;}
   \end{align*}

21) Which of the above types of code would be best to use for a program that checks a test score and assigns a letter grade? Select A, B, C or D.

22) Which of the above types of code would be best to use for a program that checks conditions for a tax program, where multiple conditions may all be true? Select A, B/C or D.

23) Which of the above types of code would be best to use for a program that checks whether input is upper-case, lower-case, or numeric? Select A, B, C or D.
24) How many errors would you need to correct in the code shown below in order to get it to compile and run correctly?

```c
char value;
int

// Underline

printf("Enter yearly salary: ");
scanf("%d", value);

// Error

if( value = 10000 ) {
    printf("You are making 10000\n");
} else
    printf("You are making something else.\n");
```

a) 3 or 4 errors  
b) 5 or 6 errors  
c) 7 or 8 errors  
d) 9 or 10 errors

25) Given the following code segment:

```c
Consider the following possible function calls that could go in place of the underline above:

I. f1( &x, &y);
II. f2( x, y);
III. f3( x, y);

Which of these options would cause the code to print the value: 9

a) I only  
b) I and II  
c) I and III  
d) II and III
```

```c
char value;

// Underline

printf("Enter yearly salary: ");
scanf("%d", value);

// Error

if( value == 10000 ) {
    printf("You are making 10000\n");
} else
    printf("You are making something else.\n");
```
26) What is the output from the code segment shown at right below, called with `scope();`

```
int x=1;  // global variable
void s2( int y)
{
    printf("%d", x);
}
void s1( int y)
{
    int x=2;
    s2( y);
}
void scope()
{
    int x=3;
    s1( x);
}
```

a) 1  
b) 2  
c) 3  
d) 4

27) What is the output from the code segment shown below, called with `s();`

```
int m1( int a, int b)
{
    if( a>b)
        return a;
    else
        return b;
}
int m2( int a, int b)
{
    if( a<b)
        return a;
    else
        return b;
}
void s()
{
    int a=5, b=2, c=6;
    int x = m2( m2( a,b), c);
    int y = m1( m1( a,b), c);
    int z = a+b+c - x - y;
    printf("%d %d %d", x, z, y);
}
```

a) 5 2 6  
b) 6 5 2  
c) 2 5 6  
d) 6 2 5

28) What is the output of calling: `f3("bookkeeper");` 

```
void f3( char text[])
{
   int lastIndex = strlen( text) - 1;
   int count = 0;
   for( char *pWord = text; pWord < &text[lastIndex]; pWord++) {
       if( *pWord == *(pWord+1) ) {
           printf("%c", *pWord);
           if a character is the same as the next one
       }
   }
}
```

a) bper    
b) boke    
c) bokeper  
d) bookkeeper
29) What is the output of the code below when called with \texttt{f2()};

\begin{verbatim}
char * f1(char theWord[], char c)
{
    int x = 0;
    int count = 0;
    for( int i=strlen( theWord)-1; i>=0; i-- ) {
        if( theWord[ i] == c ) {
            count++;
            if( count > 2) {
                return &theWord[ i];
            }
        }
    }
    return theWord;
}
void f2()
{
    printf("%s", f1("dividian", 'i') );
}
\end{verbatim}

\begin{enumerate}
    \item a) dividian
    \item b) dividian
    \item c) idian
    \item d) ian
\end{enumerate}

30) What is the output of the code shown below when called with \texttt{s3()};

\begin{verbatim}
void s2( char a[], char b[], char c[])
{
    char d[ 81];
    char *pA = strstr(a, b);
    char *pB = strchr(pA,' ');
    strcpy( d, pB);
    *pA = '\0';
    strcat( a, c);
    strcat( a, d);
}
void s3()
{
    char msg[ 81] = {"Were you ok with this test?"};
    s2( msg, "ok", "ecstatic");
    printf("%s", msg);
}
\end{verbatim}

\begin{enumerate}
    \item a) ok
    \item b) ecstatic
    \item c) Were you ok with this test?
    \item d) Were you ecstatic with this test?
\end{enumerate}