What is your name?: ___________________________(2 points. Don’t miss this one!)

There are two sections:

I. True / False 62 points; (31 questions, 2 points each)
II. Multiple Choice 36 points; (6 questions, 6 pts. each)

_____________

98 + 2 points for name = 100 points total

This test is worth 15% of your final grade. For all problems you must put your answers on the bubble form using a #2 pencil. This test is open book and open notes. You have 50 minutes.

I. True / False Section: (1 pt. each) Select the best answer to each problem below. Be careful - some are tricky.

- If the answer is True, fill in a on your answer form. If the answer is False, fill in b on your answer form.
- Several problems ask you to determine whether something is valid. Something is valid if it would not generate a compiler error and would execute without the program crashing.

T    F    1. To add 3 to the integer variable x, we could use either:
       x = x+3;
       or
       x+=3;

T    F    2. Using 10 separate integer variables to store 10 test scores in a program is just as easy as declaring an array of 10 integers.

T    F    3. If a 2-dimensional array called dictionaryWords is declared inside of main() and we want to write a function to display the words, the following could be a valid function call from within main:
       void displayWords( dictionaryWords[ NumberOfWords][ WordSize]);

T    F    4. To pass a 2-dimensional array called phraseWords to a function, the sizes of both dimensions should be specified in the function declaration, such as:
       void theFunction( char phraseWords[ NumberOfWords][ WordSize])
       Otherwise the program can go past the end of the array.

T    F    5. An if-else statement can be implemented using two if statements.

T    F    6. A do while loop combined with an if statement can implement the same functionality as a while loop.

T    F    7. A do loop can be used interchangeably with a while loop. It really comes down to user preference.

T    F    8. To ensure a character stored in variable c is upper case, we could use the following:
       if( (c>='a') && (c<='z') )
         c = c - ('A' - 'a');

T    F    9. To check if character c is a numeric character, we could use the if statement:
       if('0' <= c <= '9')

T    F    10. The break statement is used to break out of the program, terminating execution.

T    F    11. No matter what the precedence is, the expression x+++y will always have the same effect.
12. Given the declarations

```
int i;
char aWord[15];
```

If some section of code that writes to array `aWord` goes past the end of the array, it would modify variable `i`. This would not generate an error when the program is compiled.

13. Again considering the declarations given in the previous problem. As before, consider the situation if some section of code that writes to array `aWord` goes past the end of the array. This could give run-time errors sometimes, but not always.

For the next 5 problems, consider the declarations shown below.

```
char fullText[] = {"Avoid lots and lots of redundancy"};
char word2[] = {"Avoid"};
char word4[] = {"lots"};
char *pWord = word4;
```

14. The statement

```
    cout << fullText[0] << endl;
```

will display the text

```
A
```

15. The statement

```
    cout << fullText << endl;
```

has the same effect as the statement

```
    cout << word2 << endl;
```

16. The statement

```
    cout << fullText << endl;
```

has the same effect as the statement

```
    cout << &fullText[0] << endl;
```

17. The expression

```
    (strcmp( strstr(fullText, "lots") + 9, "lots") == 0)
```

evaluates to true.

18. The statements

```
pWord = strstr( fullText, "lots" );
char *pWord2 = strstr( pWord + 1, "lots" );
strcpy( pWord, pWord2 );
    cout << pWord << endl;
```

will display the text

```
Avoid lots of redundancy
```
For the following 8 problems, consider the declarations:

```
struct Employee {
    int age;
    float hourlyWage;
};
Employee anEmployee;
Employee *pEmployee = &anEmployee;
anEmployee.age = 21;
anEmployee.hourlyWage = 8;
```

T  F  19. The age of anEmployee be displayed using: `cout << age;`
T  F  20. The age of anEmployee be displayed using: `cout << anEmployee.age;`
T  F  21. The age of anEmployee be displayed using: `cout << pEmployee.age;`
T  F  22. The age of anEmployee be displayed using: `cout << *pEmployee.age;`
T  F  23. The age of anEmployee be displayed using: `cout << *(pEmployee.age);`
T  F  24. The age of anEmployee be displayed using: `cout << (*pEmployee).age;`
T  F  25. The age of anEmployee be displayed using: `cout << pEmployee->age;`
T  F  26. The age of anEmployee be displayed using: `cout << anEmployee->age;`

T  F  27. The following is a valid identifier declaration and initialization:
```
int Class = 3;
```
T  F  28. The following is a valid identifier declaration and initialization:
```
int sum = 5;
int totalSum = 6;
sum + 1 = totalSum;
```

T  F  29. The following code will give 5 numbers that are all the same.
```
for (int i=0; i<5; i++) {
    srand( 0);
    cout << rand() % 26 << endl;
}
```

T  F  30. After declaring an array dynamically using `new`, it is not necessary to release that memory explicitly since the program automatically releases it once the program terminates.

T  F  31. Consider using a bubble sort (as discussed in class) to sort 11 numbers. The maximum number of passes we need to make through the data before the numbers are sorted is 11.
II. Multiple Choice: (6 pts. each) For the following 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.

32. Consider the program given below.

```cpp
#include <iostream>
using namespace std;
int main()
{
    int x;
    int y=0;
    cout << "Please enter 10 integers: ";
    for (int i=0; i<10; i++) {
        cin >> x;
        y += x;
    }
    cout << y << "\n";
    return 0;
}
```

Its output can best be described as:

a) The largest input value
b) The sum of all the input values
c) The smallest input value
d) The average of the input values
e) None of the above

33. Consider the program given below.

```cpp
#include <iostream>
using namespace std;
int main()
{
    char letters[5];
    int i;
    for (i='A'; i<5; i++) {
        letters[ i] = i + 'a'-'A';
    }
    for (i=0; i<5; i++) {
        cout << letters[ i];
    }
    cout << endl;
    return 0;
}
```

Its output is:

a) ABCDE
b) abcd
c) EDCBA
d) edcba
e) None of the above
34. What is the output of the program given in the two columns below?

<table>
<thead>
<tr>
<th>#include &lt;iostream&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>using namespace std;</td>
</tr>
<tr>
<td>int a,b,c;</td>
</tr>
<tr>
<td>void confuse1(int b, int a)</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>a += 1;</td>
</tr>
<tr>
<td>b += 2;</td>
</tr>
<tr>
<td>c += 3;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>void confuse2(int a, int &amp;c)</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>a += 2;</td>
</tr>
<tr>
<td>b += 3;</td>
</tr>
<tr>
<td>c += 4;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>void confuse3(int &amp;a, int c)</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>a += 3;</td>
</tr>
<tr>
<td>b += 4;</td>
</tr>
<tr>
<td>c += 5;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>int main()</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>int a,c;</td>
</tr>
<tr>
<td>a=b=c=1;</td>
</tr>
<tr>
<td>confuse1( a, b);</td>
</tr>
<tr>
<td>confuse2( b, a);</td>
</tr>
<tr>
<td>confuse3( c, b);</td>
</tr>
<tr>
<td>cout &lt;&lt; a &lt;&lt; b &lt;&lt; c &lt;&lt; endl;</td>
</tr>
<tr>
<td>return 0;</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>

a) 585  
b) 415  
c) 582  
d) 412  
e) None of the above

35. Consider the following section of code:

int x, y, z;  
int *xPtr, *yPtr, **zPtr;  
x = 7; y = 5; z = 3;  
xPtr = &x;  
yPtr = &x;  
zPtr = &xPtr;

The results of the statement

    cout << **zPtr;

is:

a) 3  
b) the address in memory of x  
c) the address in memory of xPtr  
d) the address in memory of zPtr  
e) None of the above
36. What is the output of the program given below?

```cpp
#include <iostream>
#include <cstring>
using namespace std;

void theFunction( char w[])
{
    int x = strlen( w);
    char c;
    for (int i=0; i<x/2; i++) {
        c = w[i];
        w[i] = w[x-i-1];
        w[x-i-1] = c;
    }
}

int main()
{
    char array[] = {"A man a plan a canal panama");
    theFunction( array);
    cout << array << "\n";
    return 0;
}
```

Its output is:

- a) The contents of the array in reverse order
- b) The contents of the array in the original order
- c) The contents of the array with the characters shifted over by one position
- d) The contents of the array with the characters rearranged so they are no longer recognized as the words in the original phrase.
- e) None of the above

37. What is the output of the program given in the two columns below?

```cpp
#include <iostream>
using namespace std;

class DayOfYear
{
public:
    void set(int newMonth, int newDay);
private:
    int year;
    int month;
    int day;
};

void DayOfYear::set(int newMonth, int newDay)
{
    month = newMonth;
    day = newDay;
}

int main()
{
    DayOfYear today;
    today.set(12, 25);
    cout << "Today's date is ";
    cout << today.month << "/";
    cout << today.day << endl;
    return 0;
}
```

- a) 12/25
- b) 12/25/0
- c) Undetermined since the year value is not set
- d) Compiler Error: program is written incorrectly
- e) None of the above