1. Write your name on your scantron in the space labeled NAME.
2. Write CS 1613 in the space labeled SUBJECT.
3. Write the date (5-3-2004) in the space labeled DATE.
4. Write your CRN (20171) in the space labeled HOUR.
5. You may not consult your neighbors, colleagues, or fellow students to answer the questions on this test.
6. Mark all selections that satisfy the question. If selection b, c, and d satisfy a question then mark selections b, c, and d.
7. Darken your selections completely. Make a heavy black mark that completely fills your selection.
8. Answer all 50 questions.
9. You may use a hand calculator.
10. You may write on your test questionnaire.
1. What is printed by program q01?
   a. 2 8
   b. 1 5
   c. 2 4
   d. 1 3

```cpp
#include <iostream>
using namespace std;
int main()
{
  int v[5];
  for (int c=0; c<5; c++) {
    if (c<2) v[c]=c+1; else v[c]=v[c-1]+v[c-2];
  }
  return 0;
}
```

Figure 1. Program q01.

2. Mark all correct statements.
   a. The standard for C++ requires that array bounds be checked at execution time.
   b. The standard for C++ does not require that array bounds be checked at execution time.
   c. The standard for C++ requires that an array’s size declarator be a constant integer expression.
   d. The standard for C++ does not require that an array’s size declarator be a constant integer expression.

3. Which of the following programs produces a compilation error?
   a. q03a
   b. q03b
   c. q03c
   d. none of the above.

```cpp
#include <iostream>
using namespace std;
int main()
{
  return 0;
}
```

Figure 3a. Program q03a.

```cpp
#include <iostream>
using namespace std;
int main()
{
  int A[7];
  return 0;
}
```

Figure 3b. Program q03b.
#include <iostream>
using namespace std;
int main()
{   int A[7];
    return 0;
}

Figure 3c. Program q03c.

4. Which of the following programs could produce an error when executed?
   a. q03a
   b. q03b
   c. q03c
   d. none of the above.

5. Mark all syntactically correct C++ declarations.
   a. int A[]={{1,1,2,3,5,8}};
   c. char* C=\"Jan\";
   d. double D[2]=\{1.602e-19,9\};

6. One (1) byte is allocated for each character and four (4) bytes are allocated for each integer. What is printed by program q06?
   a. 4 4
   b. 4 16
   c. 5 4
   d. 5 16

#include <iostream>
using namespace std;
int main()
{   char A[]="four"; int B[]={1,2,3,4};
    cout << sizeof A << " " << sizeof B;    
    cout << endl;
    return 0;
}

Figure 6. Program q06.

7. Mark all C++ declarations that conform to the C++ language standard.
   a. char A[2,5];
   b. char B[2][5];
   c. char C[0..1,0..4];
   d. char D(2,5);

8. Assume that A is an array of characters having two (2) rows and five (5) columns. Mark all sequences of C++ expressions that reference the element in the first row and the second column.
   a. A[0][1]
   b. A[1][2]
   c. *((char*)A+1)
   d. *((char*)A+7)
9. Parameter $A$ is an array of characters passed by reference in function $F$. Mark all correct C++
declarations for parameter $A$.
   a. `void F(char A[]);`
   b. `void F(string A);`
   c. `void F(char* A);`
   d. `void F(string& A);`

10. Mark all correct statements about C++ structures.
    a. A structure is an aggregate where every member must have the same type.
    b. Every capability of a `class` can be implemented in a structure.
    c. Every member of a structure can be accessed by any function in which an instance of that structure
        is visible.
    d. Only `public` members of a structure can be accessed by functions in which an instance of a
        structure is visible.

11. Mark all correct statements about C++ structures.
    a. Members of a structure can be accessed by adding a period and the structure name after the
        member name.
    b. Members of a structure can be accessed by prefixing the structure name and a period to the
        member name.
    c. Members of a structure can be accessed by adding an arrow (`->`) and the structure name after the
        member name.
    d. Members of a structure can be accessed by prefixing the structure name and an arrow (`->`) to the
        member name.

12. A certain bank requires that a structure be defined that can be used to record a customers account number and current balance. Mark all declarations that satisfy the bank’s requirements.
    a. `struct Account { int number; double balance; };`
    b. `struct Account { string number; float balance; };`
    c. `struct Account {double number; char balance;};`
    d. `struct Account {string number; string balance;};`

13. Mark all correct statements about C++ structures.
    a. Two constructors in the same structure can have the same name.
    b. Two destructors in the same structure can have the same name.
    c. Two members in the same structure can have the same name.
    d. Both constructors and destructors have no return type.
14. Which of the following programs produce no compilation errors?
   a. q14a
   b. q14b
   c. q14c
   d. q14d

Figure 14a. Program q14a.

#include <iostream>
#include <string>
using namespace std;
struct Pie {
    string Flavor;
    Pie():Flavor("apple"){}
    Pie(string flavor):Flavor(flavor){}
};
int main()
{
    Pie Lemon("lemon");
    Pie Apple();
    return 0;
}

Figure 14b. Program q14b.

#include <iostream>
#include <string>
using namespace std;
struct Pie {
    string Flavor;
}
int main()
{
    Pie Lemon("lemon");
    return 0;
}

Figure 14c. Program q14c.
```cpp
#include <iostream>
#include <string>
using namespace std;
struct Pie {
    string Flavor;
    void ~Pie() { cout << "What, no pie nor piety?" << endl; }
};
int main() {
    Pie Apple();
    return 0;
}
```

**Figure 14d.** Program q14d.

15. Find the error in program q15.
   a. Member x has an invalid type.
   b. The structure tag-name is missing.
   c. Type name must appear after member names.
   d. none of the above.

```cpp
#include <iostream>
using namespace std;
struct {
    int x;
    float y;
};
int main() { return 0; }
```

**Figure 15.** Program q15.

16. Find the error in program q16.
   a. The type name string must be changed to char*.
   b. Function main does not have parameters that describe command line arguments.
   c. The structure tag-name is missing.
   d. Missing semicolon

```cpp
#include <iostream>
#include <string>
using namespace std;
struct Values {
    string name;
    int age;
}
int main() { return 0; }
```

**Figure 16.** Program q16.
17. Find the error in program q17.
   a. Every member must have a separate type declaration.
   b. Missing semicolon
   c. Improper structure name reference
   d. none of the above

   ```cpp
   #include <iostream>
   using namespace std;
   struct TwoVals { int a,b; }
   int main() {TwoVals.a=10; TwoVals.b=10; return 0;}
   ```

   Figure 17. Program q17.

18. Find the error in program q18.
   a. Structure V is initialized improperly.
   b. Structure V cannot be printed.
   c. The structure tag-name is missing.
   d. Missing semicolon

   ```cpp
   #include <iostream>
   using namespace std;
   struct ThreeVals { int a,b,c; }
   int main()
   { ThreeVals V={1,2,3};
     cout << V << endl;
     return 0;
   }
   ```

   Figure 18. Program q18.

19. Mark all correct C++ declarations.
   a. enum {red,green,blue};
   b. Enum color {red,green,blue};
   c. Enum {red,green,blue} color;
   d. enum color {red,green,blue};

20. Declarations in figure 20 describe a deck of cards. Variable D is an instance of a deck of cards. Mark all correct references to member pips in the thirteenth card.
   a. D[12].pips
   b. D.pips[12]
   c. D(12),pipes
   d. D.pips(12)

   ```cpp
   enum Suit {clubs,diamonds,hearts,spades};
   enum Pips {deuce,trey,four,five,six,seven,eight,nine,ten,jack,queen,king,ace};
   struct Card { Suit suit; Pips pips; };
   Card D[52];
   ```

   Figure 20. Declarations for question 20.
21. Declarations in figure 21 describe a deck of cards. Variable \( D \) is an instance of a deck of cards. Mark all correct references to member \( \text{pips} \) in the thirteenth card.
   a. \( D[12].\text{pips} \)
   b. \( D.\text{pips}[12] \)
   c. \( D(12).\text{pips} \)
   d. \( D.\text{pips}(12) \)

```
enum Suit {clubs,diamonds,hearts,spades};
enum Pips {deuce,trey,four,five,six,seven,eight,nine,ten,jack,queen,king,ace};
struct Deck {
  Suit suit[52];
  Pips pips[52];
};
Deck D[52];
```

Figure 21. Declarations for question 22.

22. Declarations in figure 22 describe the attributes of a rectangle. Mark all sequences of statements that can be placed inside function \( \text{main} \) without error.
   a. \( \text{Rectangle } R(5,6); \) \( \text{cout} << R.L << \text{endl}; \)
   b. \( \text{cout} << \text{Rectangle}(5,6) << \text{endl}; \)
   c. \( \text{Rectangle } R(5,6); \) \( \text{cout} << R.\text{Area}() << \text{endl}; \)
   d. \( \text{Rectangle } R(5,6); \) \( \text{cout} << \text{sizeof}(\text{Rectangle}) << \text{endl}; \)

```
#include <iostream>
using namespace std;
class Rectangle {
  double L,W;
public:
  Rectangle(double l=0.0,double w=0.0):L(l),W(w){}
  double Area(void){return L*W;};
  double Perimeter(void){return 2*L+2*W;}
};
int main()
{
  return 0;
}
```

Figure 22. Declarations for question 20.

23. Mark all correct selections.
   a. The indirection operator \( * \) has higher precedence than the multiplicative operator \( * \).
   b. The multiplicative operator \( * \) has higher precedence than the equality operator \( == \).
   c. The equality operator \( == \) has higher precedence than the assignment operator \( = \).
   d. The assignment operator \( = \) has higher precedence than the sequential evaluation operator \( , \).

24. Mark all correct selections.
   a. The postfix increment operator ++ has higher precedence than the prefix operator ++.
   b. The binary subtraction operator - has higher precedence than the unary negation operator -. 
   c. The equality operator \( == \) has higher precedence than the logical and operator \&\&.
   d. The relational operator < has higher precedence than the logical or operator ||.
25. Mark all correct selections.
   a. The prefix increment operator ++ is left associative.
   b. The assignment operator += is right associative.
   c. The additive operator + is right associative.
   d. The logical not operator ! is right associative.

26. Mark all correct selections.
   a. The left shift operator is <<.
   b. The inequality operator is <>.
   c. The exponentiation operator is ^.
   d. The bitwise or operator is |.

27. Which of the following is entirely composed of valid C++ identifiers?
   a. for he is a jolly good fellow
   b. The cow is of the bovine ilk One end is moo the other milk.
   c. and while she was sleeping
   d. two four six eight who do we appreciate

28. What set of numbers do real numbers not include?
   a. Whole numbers
   b. Counting numbers
   c. Complex numbers
   d. Rational numbers

29. Each rectangle below represents a single byte. Select correct valid representations for “A”

   a. 
   b. 
   c. 
   d.

30. Identify correct declarations that when compiled cause no errors or warnings.
   a. char a="a";
   b. int i=1.602E-19;
   c. bool b=0;
   d. double d=5;

31. Mark all selections whose expressions evaluate to the corresponding value in table 1

<table>
<thead>
<tr>
<th>Selection</th>
<th>Variables</th>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>int a=4,b=3,c=2,d=1;</td>
<td>a*b/c%a</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>int a=4,b=3,c=2,d=1;</td>
<td>a*b/c+1</td>
<td>6</td>
</tr>
<tr>
<td>c</td>
<td>int a=4,b=3,c=2,d=1;</td>
<td>++a*b-c--</td>
<td>15</td>
</tr>
<tr>
<td>d</td>
<td>int a=4,b=3,c=2,d=1;</td>
<td>-=d+c*-a/b++</td>
<td>-2</td>
</tr>
</tbody>
</table>
32. Mark all selections whose expressions evaluate to the corresponding value in table 2

<table>
<thead>
<tr>
<th>Selection</th>
<th>Variables</th>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>int i=4,j=3,k=2,m=1; j*=j+k</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>int i=4,j=3,k=2,m=1; j*=k+++m=5</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>int i=4,j=3,k=2,m=1; k*=m++=2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>int i=4,j=3,k=2,m=1; m/=i*=k++=3+m</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.

33. Mark all selections whose expressions evaluate to the corresponding value in table 3.

<table>
<thead>
<tr>
<th>Selection</th>
<th>Variables</th>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>int i=10,j=2,k=6; double x=3.2,y=-4.6;</td>
<td>i=j==j</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>int i=10,j=2,k=6; double x=3.2,y=-4.6;</td>
<td>10*!x==i</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>int i=10,j=2,k=6; double x=3.2,y=-4.6;</td>
<td>i+j+k==2*k</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>int i=10,j=2,k=6; double x=3.2,y=-4.6;</td>
<td>(x!=y)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.

34. Identify the flowchart in Figure 34 that best describes the flow of control in a C++ while-statement.

a.  
  ![Flowchart a](image1)

b.  
  ![Flowchart b](image2)

c.  
  ![Flowchart c](image3)

d.  
  ![Flowchart d](image4)

Figure 34. while-statement flowcharts.
35. Identify the flowchart in Figure 35 that best describes the flow of control in a C++ for-statement.

**Figure 35. for-statement flowcharts**
36. Find the value of \( \sum_{i=1}^{15} 2i \)

a. 60   
b. 120   
c. 240   
d. 480

37. Find the value of \( \sum_{r=0}^{10} 3^r \)

a. 88,573   
b. 59,049   
c. 177,146   
d. none of the above

38. What is displayed by program q38?

a. One for the money. Two for the show.   
Three to get ready and four to go.

b. One for the money.   
Two for the show.   
Three to get ready and four to go.

c. One for the money.   
Three to get ready

d. Two for the show.   
and four to go.

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

int main()
{
    for (int a=0; a<4; a=a+2) {
        switch (a+1) {
        case 1:
            cout << " One for the money.";
            break;
        case 2:
            cout << " Two for the show.";
            break;
        case 3:
            cout << " Three to get ready";
            break;
        case 4:
            cout << " and four to go.";
            break;
        }
        cout << endl;
    }
    return 0;
}
```

Figure 38. Program q38.
39. What is the present value of a sequence of monthly payments amounting to $250 over a term of twenty (20) years at 6.5 APR?
   a. $33,592.16
   b. $37,521.25
   c. $33,531.25
   d. none of the above

40. Identify valid mathematical expressions for the present value, $P$, of a sequence of equal payments, $R$, at periodic interest rate, $i$, for a term of $n$ periods.
   a. $P = \frac{Ri}{1 - (1 + i)^{-n}}$
   b. $P = \sum_{k=1}^{n} \frac{R}{(1 + i)^k}$
   c. $P = R \frac{1 - (1 + i)^{-n}}{i}$
   d. none of the above

41. What is displayed by program q41.
   a. kt
   b. ku
   c. lt
   d. lu

#include <iostream>
using namespace std;
void P(char c) { c++; }
void Q(char& d) { d++; }
int main()
{   char a='k',b='t';
    P(a); Q(a); P(b); Q(b);
    cout << a << b << endl;
    return 0;
}

Figure 43. Program q41.

42. Identify parameters in program q41.
   a. a
   b. b
   c. c
   d. d

43. Identify arguments program q41.
   a. a
   b. b
   c. c
   d. d
44. Mark all equalities.
   a. $11181_{10} = 2BAD_{16}$
   b. $11129_{10} = CAT_{30}$
   c. $41377 = A1A1_{16}$
   d. $6030271 = FINAL_{25}$

45. Mark all correct statements about makefiles.
   a. Comments begin with an exclamation point (!) and end with the newline character.
   b. Comments begin with two forward slash characters in succession (//) and end with the newline character.
   c. Comments begin with the two-character sequence /* and terminate with the two-character sequence */.
   d. Comments begin with pound-sign character (#) and end with the newline character.

46. The safest environment to create a makefile is the Linux/Unix environment because
   a. Only the Linux/Unix environment has the make utility.
   b. The tab-character is not reliably transferred from other environments to the Linux/Unix environment.
   c. Makefiles are not used in Windows ® and other environments
   d. none of the above

47. Parameters
   a. are defined between enclosing parentheses where the function is defined.
   b. are defined between enclosing parentheses where the function is called.
   c. specify how arguments are passed.
   d. include a type specification.

48. An argument
   a. can be passed by-value if it is an expression.
   b. can be passed by-reference if it is an expression.
   c. can be passed by value if it is a variable.
   d. can be passed by reference if it is a variable.

49. A parameter
   a. becomes an alias when the corresponding argument is passed by-value.
   b. becomes an alias when the corresponding argument is passed by-reference.
   c. is a copy of the corresponding argument when it is passed by-value.
   d. is a copy of the corresponding argument when it is passed by-reference.

50. Preprocessor directives begin with
   a. #
   b. !
   c. <
   d. None of the above