FORM 2 (put name, form, and section number on scantron!!!)
CS 162 Exam II
True (A) / False (B) (2 pts)
1. A template class allows the class to be used with different data types.  
   - T

2. Definitions of member functions cannot be placed inside the class definition.  
   - F

3. A friend function must have two or more operands from the class being defined.  
   - F

4. You can create a non-member function with one parameter that overloads the relational equal to (==) operator.  
   - F

5. Nodes in a linked list are stored in contiguous memory.  
   - F

6. The template prefix can be written template <typename identifier> or template <class identifier> with the same results.  
   - T

7. C++ enables you to use the friend keyword to declare friend functions and friend classes for a class so these functions and classes can access the class’s private members.  
   - T

8. If a function is declared as returnType functionName(parameterList) throw (type), this function can only throw the exception of the specified type.  
   - F

9. Templates are an example of algorithm abstraction.  
   - T

10. Static binding occurs when the compiler binds a function call at compile time.  
    - T

11. Pointers to a base class may be assigned the address of a derived class object.  
    - T

12. In C++ polymorphism is very difficult to achieve unless you also use inheritance.  
    - F

13. A reason to overload the square brackets [] operator is to write classes that have array-like behavior.  
    - T

14. Private inheritance will make all public members private in the derived class but leave protected members as protected in the derived class.  
    - F

15. The standard template library (STL) vector and list classes are examples of sequential containers.  
    - T

16. If an exception is not caught, it is stored for later use.  
    - F

17. Inheritance is an example of a “has a” relationship in OOP.  
    - F

Multiple Choice (3 pts):
18. Who can access private data in a class?
   A. classes derived from the class  
   B. friends of the class  
   C. everyone  
   D. A and B  
   E. no one  
   - C
19. Variables that are shared by every instance of a class are _________.
   A. instance variables  
   B. private variables  
   C. public variables  
   D. static variables

20. What is a possible correct signature for the overloaded subscript operator []?
   A. &long operator[](const int &index);  
   B. long operator[](const int &index);  
   C. long operator&[](const int &index);  
   D. long Rational::operator[](const int &index)

21. Which of the following statements are true?
   A. A custom exception class must always be derived from class exception.  
   B. A custom exception class must always be derived from a derived class of class exception.  
   C. A custom exception class is just like a regular class in C++.  
   D. A custom exception class must always be derived from class runtime_error.

22. Which of the following is a pure virtual function?
   A. virtual double getArea();  
   B. double getArea() = 0;  
   C. virtual double getArea() = 0;  
   D. virtual double getArea() { }

23. Which of the following is a FIFO (First-in-first-out) structure?
   A. Linked list  
   B. Two-way list  
   C. Stack  
   D. Queue

24. What is wrong in the following code?
   ```
   #include <iostream>
   #include <vector>
   using namespace std;
   int main()
   {
       vector<int> v;
       cout << v[0];
       return 0;
   }
   ```
   A. The program has a runtime error on vector<int> v.  
   B. The program has a runtime error on v[0], because the vector is empty.  
   C. The program has a syntax error on vector<int> v.  
   D. The program has a syntax error on v[0].

25. To add an int value 5 to a vector v of integers, use _________.
   A. v.append(5);  
   B. v.insert(5);  
   C. v.add(5);  
   D. v.push_back(5);
26. Given the following templated function:

```cpp
template<class T>
T maxValue(const T value1, const T value2)
{
    if (value1 > value2)
        return value1;
    else
        return value2;
}
```

Which of the following statements is not correct?
A. `cout << maxValue(1.5, 2)`
B. `cout << maxValue('A', 'B')`
C. `cout << maxValue(1, 2)`
D. `cout << maxValue("AB", "AB")`
E. `cout << maxValue(1.5, 2.5)`

27. A template class called example has a member function whose prototype is:

```cpp
T doesSomething (const T &);
```

Which of the following is the correct header for the implementation of doesSomething?
A. `T example::doesSomething(const T &thing)`
B. `T example<T>::doesSomething(const T &thing)`
C. `<T> example::doesSomething(const T &thing)`
D. `<T> example::doesSomething(const <T> &thing)`

28. If a function in the definition of the class entry has prototype:

```cpp
friend int myFunction (const entry &, const entry &);
```

which of the following could be a correct function call? (a, b, c are of type entry.)
A. `cout << a.myFunction (b, c);`
B. `cout << myFunction (a, b);`
C. `cout << a.myFunction (b);`
D. `cout << friend.myFunction (a, b);`

29. Suppose the rule of the party is that the participants who arrive later will leave earlier. Which data structure is appropriate to store the participants?
A. Array
B. Queue
C. Stack
D. Linked List

30. Which of the following will insert the node pointed to by newPtr between the nodes pointed to by aPtr and bPtr in a linked list?
A. `newPtr->next = aPtr;`
   `bPtr->next = aPtr;`
B. `aPtr = newPtr->next;`
   `bPtr = newPtr->next;`
C. `aPtr->next = newPtr->next;`
   `newPtr->next = bPtr->next;`
D. `aPtr->next = newPtr;`
   `newPtr->next = bPtr;`
31. Given the linked list above, and if the data field of the above nodes is called data and the pointer field is called next, which of the following is not a correct way to reference the data field of node C.

A. s->data
B. r->next->data
C. q->next->next->data
D. p->next->next->next->data

32. Which of the following two versions of the class declarations is correct?

Version I:
```cpp
class A {
private:
    A* a;
    int i;
};
```

Version II:
```cpp
class A {
private:
    A a;
    int i;
};
```

A. Both versions are correct.
B. Both versions are wrong.
C. Version II is correct.
D. Version I is correct.

Given the following classes, answer #33 and #34.
```cpp
class B {
public:
    ~B() {
        cout << "B";
    }
};
class A: public B {
public:
    ~A() {
        cout << "A";
    }
};
```

33. What is the output of the following code?
```cpp
int main() { 
    A a; 
    return 0;
}
```

A. A
B. AA
C. BA
D. B
E. AB
34. What is the output of the following code?
```c++
int main() {
    B *a = new A;
    delete a;
    return 0;
}
```
A. A  
B. AA  
C. BA  
D. B  
E. AB

35. Suppose Circle and Rectangle classes are derived from GeometricObject and you declared
```c++
void displayGeometricObject(GeometricObject shape) {
    cout << shape.toString() << endl;
}
```
Which of the following function call is incorrect?
A. displayGeometricObject(Rectangle(2, 3));
B. displayGeometricObject(GeometricObject("black", true));
C. displayGeometricObject(string());
D. displayGeometricObject(Circle(5));

36. Suppose that statement2 throws an exception of type Exception2 in the following statement:
```c++
try {
    statement1;
    statement2;
    statement3;
} catch (Exception1 ex1) {
} catch (Exception2 ex2) {
} catch (Exception3 ex3) {
    statement4;
}
statement5;
```
Which statements are executed after statement2 is executed?
A. statement2  
B. statement5  
C. statement1  
D. statement4  
E. statement3

37. Suppose we have the following definition:
```c++
vector<int> vec;
// use push_back to put 10 values into vec here.
vector<int>::iterator itr1, itr2, itr3;
itr1 = vec.begin();
itr2 = vec.begin() + 5;
```
For this iterator which of the following is incorrect?
A) *itr1  
B) itr2[3]  
C) itr2 + 5  
D) B and C
38. How many type parameters may a function template have?
   A. none, that is not what the parameters in a function template are called.
   B. 1
   C. 2
   D. as many as are needed

39. What is the printout of the following code?
   ```cpp
class C {
   public:
      string toString() {
         return "C";
      }
};
class B: public C {
   string toString() {
      return "B";
   }
};
class A: public B {
   virtual string toString() {
      return "A";
   }
};
void displayObject(C *p) {
   cout << p->toString();
}
int main() {
   displayObject(&A());
   displayObject(&B());
   displayObject(&C());
   return 0;
}
```
   A. BBB
   B. CBA
   C. CCC
   D. AAA
   E. ABC

**Extra Credit (2 pts)**
40. True(A)/False(B) If A is derived from B, and B is derived from C, and B has a virtual function. This function is dynamically bound to a C pointer.
41. True(A)/False(B) In the template prefix, template<class T> the keyword class means that the type parameter T must be of class type
42. True(A)/False(B) A class that has a pure virtual member function is called a concrete base class.
43. True(A)/False(B) Virtual functions are implemented with a table look up that is done at run time.
44. True(A)/False(B) When an exception is thrown, the function finishes its execution, its value is returned, and then control is transferred to the catch block.