FORM 1(put name, form, and section number on scantron!!!)

CS 162 Exam II

True (A) / False (B) (2 pts)

1. The following code declares a vector of characters.
```
    vector characters<char>
```

2. The linked list is always the most efficient storage mechanism for storing many pieces of data.

3. The stack can be compared to a line of people at a bank, where the first person in line is the first person served.

4. A stack is a specialized type of list.

5. If head is a pointer to the first node in a linked list, then `*head.item` is the same as `(*head).item`.

6. Vectors and arrays are the same data type.

7. If we use an out of range index with a vector, there will be an error message from the compiler.

8. A vector v will automatically increase the allocated size when more than `v.size( )` elements are inserted with `v.push_back( newElement).`

9. Destructors are not inherited into the derived class.

10. The assignment operator is inherited from the base class.

11. If a function throws an exception, it must be caught inside that function.

12. In a try block, the throw statement is always executed.

13. The throw statement passes a value to the catch block.

14. The catch block is the group of statements that handle an exception.

Multiple Choice (3 pts):

15. The pointer in a node points to
    a. the data part of a node
    b. the count part of a node
    c. the pointer part of the node
    d. the whole node

16. Which type of exception is thrown if a call to the new operator fails?
    a. ArithmeticError
    b. DivideByZero
    c. bad_alloc
    d. MemoryError

17. In a linked list, the pointer variable head
    a. is the first node in the list
    b. points to the first node in the list
    c. is always NULL
    d. is undefined

18. Given the following declarations, which statement would put the value of 3 in the item part of the first node in the linked list?

```
struct Node {
     int item;
     Node *link;
};
Node  *head;
head = new Node;
```

    a. head=3;
    b. head.item=3;
    c. *head.item=3;
    d. head->item=3;

19. What is wrong with the following code to insert a node at the front of the list?

```
struct Node {
     int item;
     Node *link;
};
Node *head=NULL, *tmp;
//inserting a new node
tmp=new Node;
tmp->item = data;
head = tmp;
tmp->next = head->next;
```

    a. head->next is pointing to NULL
    b. if there were any nodes in the list, they are now lost.
    c. nothing is wrong.
    d. tmp should be declared to be a Node not a Node *

20. What is the value of `numbers.size()` after the following code?

```
vector<float> numbers;
numbers.reserve(100);
```

    a. 0
    b. 10
    c. 100
    d. unknown

21. To add an element to a vector of integers named numbers at the next available position in the vector, you would use:
    a. numbers[numbers.size()+1] = newValue;
    b. numbers = newValue;
    c. numbers.pushBack(newValue);
    d. numbers.push_back(newValue);

22. Which is the correct way to tell the compiler that the class being declared (ChildClass) is derived from the base class (BaseClass)?
    a. class ChildClass::public BaseClass
    b. class ChildClass:public BaseClass
    c. class ChildClass childOf public BaseClass
    d. class ChildClass derived BaseClass

23. If the member variables in a base class are private, then
    a. they can be directly accessed or changed in the derived class
    b. the derived class must use any accessor or modifier functions from the base class
    c. making them private causes a syntax error.
    d. you must declare them in the derived class also.

24. Give a base class with at least one public member function, how many child classes can redefine that member function?
    a. 1
    b. 0
    c. all of them
    d. none of the above

25. If the member variables in the base class are listed as protected, then who can access or modify those variables?
    a. members of the base class
    b. members of the derived class
    c. outside the base or derived classes
    d. A and B
    e. All of the above

26. If a base class has public member functions that are not listed by a derived class, then these functions
    a. are not available to the derived class
    b. are inherited unchanged in the derived class
    c. are private to the derived class
    d. do not exist in the derived class

27. If you have a copy constructor in the base class, but do not have a copy constructor for the derived class, then
    a. you will have a syntax error
    b. a copy constructor for the derived class is automatically created for you
    c. you cannot use pointer variables
    d. the default constructor is used

28. Given a class A that derives from a class B that derives from a class C, when an object of class A goes out of scope, in which order are the destructors called?
    a. C, B, then A
    b. A, B, then C
    c. unable to determine
    d. depends on how the code is written for the destructors

Questions #29-#32 Use the following classes to answer #29-32:

```
class Pet {
public:
    virtual void print();
    string name;
};

class Dog: public Pet {
public:
    void print();
    string breed;
};

void Pet::print() {
    cout << "My name is " << name;
}

void Dog::print() {
    Pet::print();
    cout << ", and my breed is a "<< breed << endl;
}
```

29. Given the following code (using the above classes),

```
Dog vDog;
Pet vPet;
vDog.name="rover";
vDog.breed = "Collie";
```

Which of the following statements are not legal?

a. vPet=vDog; cout << vDog.name;
b. vPet=vDog; cout << vDog.breed;
c. vPet=vDog; cout << vPet.name;
d. vPet=vDog; cout << vPet.breed;

30. Given the following code (using the above classes), what is the output of the last statement shown?

```
Pet* pPtr;
Dog* dPtr=new Dog;
dPtr->name= "Rover";
dPtr->breed="Weiner";
pPtr= dPtr;
pPtr->print();
```

a. My name is Rover, and my breed is a Weiner
b. My name is Rover
c. , and my breed is a Weiner
d. nothing

31. If the Pet class had a non-virtual member function named print, and a pointer variable of that class is pointing to a Dog object, then the code pPtr->print( ); calls

a. the base class print function
b. the derived print function
c. both the derived and base print functions
d. it causes a run-time error

32. Given the following code (using the above classes), what is the output of the last statement shown?

```
Pet pPtr;
Dog dPtr;
dPtr.name= "Rover";
dPtr.breed="Weiner";
pPtr= dPtr;
pPtr.print();
```

a. My name is Rover, and my breed is a Weiner
b. My name is Rover
c. , and my breed is a Weiner
d. nothing

33. Polymorphism refers to

a. the ability to change the behavior of a function at runtime.
b. overriding base class functions.
c. overloading functions
d. none of the above

34. In order to tell the compiler to wait to decide which version of a function to use, you must precede the function declaration in the base class with the keyword

a. operator
b. friend
c. virtual

35. Which of the following should be virtual if a base class uses dynamic memory allocation?

a. the constructor
b. the copy constructor
c. the print function
d. the destructor

36. You should make a function a virtual function if

a. every class that is derived from this class use all the member functions from this class.
b. every class that is derived from this class needs to re-define this function.
c. that function is an operator
d. only in the derived classes

37. Which of the following operations do forward iterators have?

a. Overloaded `operator+` to add an `int` value to the iterator to move the place the iterator points forward by the argument number of elements.
b. Overloaded `operator*` to multiply the iterator by an `int` value to move the place the iterator points by a number of elements equal to the argument.
c. Overloaded `operator++` to move the place the iterator points forward by one element.
d. Overloaded `operator--` to move the place the iterator points backward by one element.

38. I have an algorithm that runs in $O(n^2)$ time, where n is the size of the problem. What does “the size of the problem” mean?

a. The size of the problem is the number of bytes the data occupies
b. The size of the problem is the number of lines in the source code of the program.
c. The size of a problem is the number of data items that the algorithm operates upon
d. The size of the problem is the depth of nesting of loops in the program.

Extra Credit (2 pts)

39. True(A)/False(B) Templates are an example of algorithm abstraction

40. Suppose we have the following definition:

```
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;
itr3 = vec.end();
```

For this iterator which of the following is incorrect?

A) `*iter1`
B) `itr2[3]`
C) `itr3 + 3`
D) `itr2 - 5`

41. I have an algorithm that runs in $O(N^2)$, where N is the size of the problem. For N = 100, the time for the algorithm to run is 1 minute. How long does the algorithm take for N=1000?

A) Same time
B) 10 minutes
C) 100 minutes
D) 1000 minutes

42. True(A)/False(B) Friend functions are members of the class.

43. Who can access private data in a class?

A) classes derived from the class
B) friends of the class
C) everyone
D) B and C
E) no one