

# CS 162 Exam II Spring 2018 FORM 1

Key

Please put your name and form number on the scantron.

## True (A)/False (B) (28 pts, 2 pts each)

- 1. All operators can be overloaded.
- 2. You cannot create new operators (such as the quote).
- 3. If you use the keyword virtual in a function declaration, you must also use it in the function definition.
- 4. The destructor is inherited from the base class.
- 5. All member functions in a base class should be listed as virtual functions.
- 6. The derived class may define variables and member functions other than those that are in the base class.
- 7. If the member variables in a base class are marked as private, the derived class directly access those variables.
- 8. A base class pointer variable can point to derived class objects only.
- 9. If a function throws an exception, it must be caught inside that function.
- 10. In a try block, the throw statement is always executed.
- 11. The catch block is the group of statements that handle an exception.
- 12. If a throw list has multiple exceptions listed, they are separated by semicolons.
- 13. A class that has a pure virtual member function is called a concrete base class.
- 14. If your program defines a class template, then the compiler will generate a class for each different data type for which it is instantiated.

## Multiple Choice (72 pts, 3 pts each)

- 15. A \_\_\_\_\_ function is not a member of the class, but has access to the private members of the class.
  - a. Parent
  - b. Child
  - c. Friend
  - d. Leech
  
- 16. Operators can be overloaded as
  - a. friends of a class
  - b. members of a class
  - c. non-friends, non-members of a class
  - d. All of the above
  
- 17. If the member variables of the base class are marked as protected, who can access those variables?
  - a. Users
  - b. Other programs
  - c. Derived classes
  - d. All of the above

18. The ability to associate multiple meanings to one function name using dynamic binding is called

- a. Inheritance
- b. Object Oriented
- c. Polymorphism
- d. Inconsistency

19. A base class may have at most \_\_\_\_\_ child class derived from it.

- a. 1
- b. 2
- c. 12
- d. any number

20. 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`

21. In the derived class definition, you list from the base class

- a. all the member functions every time
- b. only those member functions that need to be redefined
- c. only those member functions that were in the public section
- d. only those member functions you want to overload.

22. Using inheritance allows us to

- a. eliminate duplicate code
- b. make our classes more modular
- c. use polymorphism
- d. all of the above
- e. none of the above

23. Which of the following are not true

- a. an object of the derived class may be stored in a variable of the base class
- b. an object of the base class may be stored in a variable of the derived class
- c. an object of a derived class that is derived from another class that is derived from a third class can be stored in a variable of the third class.
- d. all of the above
- e. none of the above

24. 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 can not use pointer variables
- d. the default constructor is used

25. Which of the following would correctly call the base class (BaseClass) assignment operator from the derived class (DerivedClass) assignment operator?

```
DerivedClass& DerivedClass::operator =(const DerivedClass& rightSide)
{
    //what goes here?
}
```

- a. BaseClass::operator=(rightSide);
  - b. leftSide=rightSide;
  - c. rightSide=BaseClass.rightSide;
  - d. DerivedClass::rightSide=BaseClass::rightSide;
26. If the following function is in a base class, which of the following are polymorphic declarations of the same function in the derived class?  
virtual void print( ostream& out);
- a. virtual void print ( ostream& out);
  - b. void print( ostream& out);
  - c. void print();
  - d. virtual void print();
  - e. A and B
27. If a base class has a non-virtual member function named print, and a pointer variable of that class is pointing to a derived object, then the code ptr->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
28. The catch block  
catch (...)  
is known as the \_\_\_\_\_.
- a. Default catch block
  - b. A syntax error
  - c. The ellipses block
  - d. None of the above
29. If the following function throws an unhandled exception, what happens?  
void f1() throw ( );
- a. The program will exit
  - b. the exception will propagate to the calling program or function
  - c. The program will hang
  - d. None of the above

30. If you define a function in the derived class that has the same function signature as a function in the base class, this is known as
- overloading
  - redefinition
  - overwriting
  - a syntax error
31. A catch block that expects an integer argument will catch
- all exceptions
  - all integer exceptions
  - any exception value that can be coerced into an integer
  - none of the above
32. If you want to make a function a template function, you must precede the function declaration and definition with
- template
  - template <class int>
  - template <class T>
  - template <void>
33. Which of the following describes a class that would be a good candidate for conversion to a template class?
- A class which defines a new type of array
  - A class which defines rational numbers
  - A class which defines customers for a store
  - All of the above
34. If you define some list class template in your program, and then declare a list of integers, 2 lists of doubles and 1 list of strings, how many different version of the template class will the compiler provide?
- 1
  - 2
  - 3
  - 4
35. Which of the following is an incorrect declarations of iterators for STL containers? You may assume that the proper header has been included and that a using directive makes the names from namespace std available.
- vector<int>::iterator vecIterator;
  - list::iterator<int> listIterator;
  - deque<int>::iterator dequeIterator;
  - list<int>::iterator listIterator;
36. To add an element to a vector of integers named numbers at the next available position in the vector, you would use:
- numbers[numbers.size()+1] = newValue;
  - numbers = newValue;
  - numbers.pushBack(newValue);
  - numbers.push\_back(newValue);



37. Which of the following operations do forward iterators have?
- Overloaded operator+ to add an int value to the iterator to move the place the iterator points forward by the argument number of elements.
  - 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.
  - Overloaded operator++ to move the place the iterator points forward by one element.
  - Overloaded operator-- to move the place the iterator points backward by one element.

38. 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;
```

- head=3;
- head.item=3;
- \*head.item=3;
- head->item=3;

### Extra Credit (2 pts each)

39. Given the function below, which of the following are needed to change the function into a function template?

```
int smallest( int array[], int size)  
{  
    int small=0, i;  
    for(i=0;i<size;i++)  
    {  
        if(array[i] < array[small])  
            small=i;  
    }  
    return small;  
}
```

- precede the function definition with template <class T>
- change the type of the array to T
- change all occurrences of int to T
- All of the above
- A and B

40. True(A)/False(B): The STL containers each define iterators appropriate to the internal structure of the container.

41. In Linked Lists, 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

42. We generally set a pointer variable to NULL

- a. to signify that the pointer does not point to any memory
- b. because we want all our pointers to always point to NULL
- c. never

43. Given the following search function declaration, what would be the corresponding declaration for a templated search function?

```
int search( int array[], int start, int target, int size);
```

```
//pre: start is > 0, and < size
```

```
//the position of the first occurrence of target at or after start is returned, or -1 is returned.
```

- a. 

```
template <class T>
int search(int array[], int start, T target, int size);
```
- b. 

```
template <class T>
T search(T array[], T start, T target, T size);
```
- c. 

```
template <class T>
int search(T array[], int start, T target, int size);
```
- d. 

```
template <class T>
int search(T array[], T start, T target, T size);
```
- e. all of the above