

Name: _____

Student ID number: _____

CS 14 – Midterm
100 points possible

True/False questions. Please mark A for True and B for False on the answer sheet provided:

(3 pts each)

1. Anything that can be solved by using queues can be solved by using stacks.

2. A stack is a FIFO structure.

3. Given the following class definition:

```
class DerivedClass : inheritanceType BaseClass
```

If you use *public inheritance*, public and protected members of the base class are private members of the derived class

4. Given the following class definition:

```
class DerivedClass : inheritanceType BaseClass
```

When using inheritance, the derived class constructor is the first constructor to be called.

5. A queue is a FIFO structure

6. Friend functions can access the public and protected members of a class, but not the private members of the class.

7. Templates can have more than one data-type parameter.

8. An overloaded operator is an example of polymorphism.

9. $*(anArray+1)$ is equivalent to $anArray[1]$

Multiple choice questions. Please mark the answer on the answer sheet provided:
(3 pts each)

10. A doubly linked list is declared as:

```
class Node {  
    public:  
        itemtype item;  
        Node* next;  
        Node* prev;  
};  
  
class List {  
    private:  
        Node* head;  
};
```

Which of the following segments of code removes the element pointed to by X from the doubly linked list, if it is assumed that X points to neither the first nor the last element in the list?

- A. `X->prev->next = X->next;`
`X->next->prev = X->prev;`
- B. `X->prev->next = X->prev;`
`X->next->prev = X->next;`
- C. `X->prev->prev = X->next;`
`X->next->next = X->prev;`
- D. `X->prev->prev = X->prev;`
`X->next->next = X->next;`
- E. `X->prev = X->next;`
`X->next = X->prev;`

11. Which of the following is the correct postfix notation given the infix expression:

$((a + b) * (c + d)) - (e * f)$

- A. `abcd+*+ef*-`
- B. `ab+cd+ef*-*`
- C. `ab+cd+*ef*-`
- D. `ab+cd*ef*-+`
- E. None of the above

12. Lists may be either array-based or pointer-based (linked lists). Which of the following specify an **advantage** of a **pointer-based** implementation of a list over an array-based implementation of a list:

- A. Less space for each item
- B. Faster inserts and removes in the middle of the list
- C. No unused space
- D. A and B
- E. B and C

13. Lists may be either array-based or pointer-based (linked lists). Which of the following specify an **advantage** of an **array-based** implementation of a list over a pointer-based implementation of a list:

- A. Less space for each item
- B. Faster inserts and removes in the middle of the list
- C. No unused space
- D. A and B
- E. B and C

14. Which Abstract Data Type (ADT) would be **best** for an operator answering phone calls:

- A. Stack
- B. Queue
- C. List

15. Which Abstract Data Type (ADT) would be **best** for a word processor's implementation of the Undo button (multiple undos are allowed):

- A. Stack
- B. Queue
- C. List

16. Which of the following **best** describes/exhibits encapsulation:

- A. Overloading
- B. Classes
- C. Polymorphism
- D. All of the above
- E. None of the above

Consider the following class definitions:

```
class Sphere {  
public:  
    virtual double getArea ();  
    void displayStatistics ();  
};
```

```
class Ball: public Sphere {  
public:  
    double getArea ();  
    void displayStatistics ();  
};
```

Suppose that the implementation of each version of `displayStatistics` invokes the function `getArea`. Given the statements:

```
Ball myBall;  
SpherePtr = &myBall;
```

17. Which version of `displayStatistics` will the call `spherePtr->displayStatistics()` invoke?

- A) the Sphere class `displayStatistics` member function
- B) the Ball class `displayStatistics` member function

18. Which version of `getArea` will the call `spherePtr->displayStatistics()` invoke?

- A) the Sphere class `getArea` member function
- B) the Ball class `getArea` member function

Short Answer.

19. (5 pts) Briefly define Abstract Data Type (ADT).

20. (5 pts) Briefly define object oriented programming (OOP).

21. (8 pts) List **four** advantages when using a modular design.

22. (4 pts) Given the following code snippet:

```
int* x;  
int a = 0;  
x = &a;  
*x = 5;
```

Describe what each of these statements will print:

a) `cout << x << endl;`

b) `cout << &x << endl;`

c) `cout << *x << endl;`

d) `cout << a << endl;`

23. (6 pts) Write a C++ implementation of a template function that returns the maximum of two variables of the template type.

24. (8 pts) Write a function to reverse the elements in an array given the following function prototype. You must reverse the array in place, meaning that you may not use a second array. Remember to use good programming style.

```
void reverse ( itemtype array[], int arraySize );
```

25. (10 pts) Write a function to reverse the nodes in a **singly linked list with only a pointer to the head of the list (no last pointer)** and no sentinel (dummy) node. Your function should be general in that it would work for any item type. **You must reverse the nodes of the list via pointer manipulations. You may not simply copy the item from one node to the next.** Remember to use good programming style. You may assume the following classes exist:

```
class List {  
private:  
    Node* head;  
public:  
    void reverse();  
};
```

```
class Node {  
public:  
    itemtype item;  
    Node* next;  
};
```