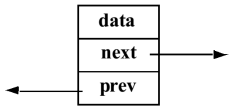
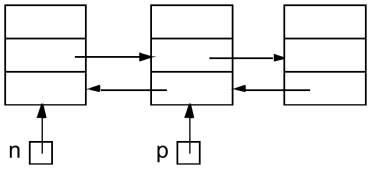
CS003A **Osbaldo Gonzalez Jr.**
Final Exam Review

1. Nodes for a doubly linked list are defined to have the following structure:



The next instance variable stores a reference to the next node in the list, and the prev instance variable refers to the previous node in the list. Below is a list of three of these nodes, along with two reference variables, n and p, that refer to specific nodes in the list.



//since they are references, the pointer is "hidden" and we just use them more or less as normal variables

Which of the following expressions does not refer to the third node in the list?

a. p.next //third node
b. n.next.next //third node
c. **p.prev.next** //**second node**
d. p.next.prev.next //third node
e. n.next.next.prev.next //thirdnode

2. If class D is a derived class that inherits from class B, which of the following occurs when a D object is constructed?
   a. A constructor of the D class is executed and then a constructor of the B class is executed.
   b. **A constructor of the B class is executed and then a constructor of the D class is executed.**
   c. A constructor of the D class is executed and a constructor of the B class is not executed.
   d. A constructor of the B class is executed and a constructor of the D class is not executed.
   e. A constructor of the D class is executed and then a constructor of the B class is executed only if it is explicitly called by the D class constructor.
   f. A constructor of the D class is executed only if it is declared virtual. Otherwise, a constructor of the B class is the only constructor executed.

3. Given the same classes B and D as in the previous question, which of the following statements has/have a syntax error? (Assume both B and D have default constructors.)
   a. B * b = new B;
   b. D * d = new D;
   c. B * d = new D; **//this is fine since D is derived from class B**
   d. D * b = new B;
   e. c) and d) have syntax errors
   f. none of the above statements have syntax errors.

4. Dynamic binding refers to the process of _____.
   a. giving the compiler specific instructions as to which function to use
   b. **delaying until run-time the choice of the appropriate function to use**
   c. using the scope resolution operator to decide which function to use
   d. using the keyword new to allocate memory for the function code

5. Circle the correct answer

```
struct ListNode
{
   float     volume;
   ListNode* link;
};
ListNode* headPtr;
ListNode* ptr;
float     searchVal;
```

Assume that headPtr is the external pointer to a linked list of many nodes. Which code segment below searches the list for the first occurrence of searchVal, leaving ptr pointing to the node where it was found? (Assume searchVal is definitely in the list.)

a.
```
ptr = headPtr;
while (volume != searchVal)
    ptr = link;
```

b.
```
ptr = headPtr;
while (ptr.volume != searchVal)
    ptr = ptr.link;
```

c.
```
ptr = headPtr;
while (ptr->volume != searchVal)
    ptr++;
```

d.
```
ptr = headPtr;
while (ptr->volume != searchVal) //this is the answer
    ptr = ptr->link;
```

e.
```
ptr = headPtr->volume;
while (ptr != searchVal)
    ptr = ptr->link;
```

6. To indicate that a member function of a class is pure virtual
   A) you must put = 0 where the body of the function would go. **//you must also use the virtual keyword**
   B) you must include the <pure> header file.
   C) you use the keywords pure virtual
   D) All of the above
   E) None of the above

7. What is the output of the following program?

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

int main(){
     try{
          try{
          throw 20;
      }
      catch (int n){
          cout << "Inner Catch\n";
          throw;
      }
     }
     catch (int x){
          cout << "Outer Catch\n";
     }
     return 0;
}
```

Output:

Inner Catch

Outer Catch

8. Which function gets called in the following code?

```
class Person
{
  public: virtual void print( ) const;
};
class Student : public Person
{
  public: void print( ) const;
};

int main( ) {
  Person *x = new Student( );
  x->print( );
  return 0;
}
```

a. `Person::print()`
b. **`Student::print()`**
c. `void print()`
d. No function gets called, the call results in a segmentation fault

9. Given an integer $n > 0$, write a recursive C++ function that returns the sum of the squares of 1 through n.

```
long sumOfSquares(long n)

{

     if (n == 1)

          return 1;
```

```
        return (n*n) + sumOfSquares(n - 1);

}
```

10. What is the output of the following program.

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

template <class T>
class Test
{
private:
    T val;
public:
    static int count;
    Test()  {   count++;   }
};

template<class T>
int Test<T>::count = 0;

int main()
{
    Test<int> a;
    Test<int> b;
    Test<double> c;
    cout << Test<int>::count   << endl;
    cout << Test<double>::count << endl;
    return 0;
}
```

Output:

2

1

11. Suppose a program has the following class declarations:

```
class Shape
{
private:
       double area;
public:
       void setArea(double a)
              {      area = a;}
       double getArea()
              {      return area;}
};

class Circle: public Shape
{
private:
       double radius;
public:
       void setRadius(double r)
       {
              radius = r;
              setArea( 3.14 * r * r);
       }

       double getRadius()
       {      return radius; }
};
```

Answer the following questions concerning these classes:

A) When an object of the `Circle` class is created, what are its `private` members?
double radius;

B) When an object of the `Circle` class is created, what are its `public` members?
void setArea(double);
double getArea();
void setRadius(double);
double getRadius();

C) What members of the `Shape` class are not accessible to member functions of the `Circle` class?
Any members that are defined as private. Protected and Public members are accessible to the circle class always (assuming public inheritance)

12. Suppose a program has the following class declaration:

```
// Declaration of CheckPoint class.
Class CheckPoint
{
   private:
         int a;
   protected:
         int b;
         int c;
         void setA(int x) { a = x;}
   public:
         void setB( int y) { b = y;}
         void setC( int z) { c = z;}
};
```

Answer the following questions regarding the class:

A) Suppose another class, `Quiz`, is derived from the `CheckPoint` class. Here is the first line of its declaration:

```
class Quiz : private CheckPoint
```

Indicate whether each member of the `CheckPoint` class is `private`, `protected`, `public`, or inaccessible:

```
a          //inaccessible
b          //private
c          //private
setA       //private
setB       //private
setC       //private
```

B) Suppose the `Quiz` class, derived from the `CheckPoint` class, is declared as class `Quiz : protected CheckPoint`
Indicate whether each member of the `CheckPoint` class is `private`, `protected`, `public`, or inaccessible:

```
a          //inaccessible
b          //protected
c          //protected
setA       //protected
setB       //protected
setC       //protected
```

C) Suppose the `Quiz` class, derived from the `CheckPoint` class, is declared as class `Quiz : public CheckPoint`

Indicate whether each member of the `CheckPoint` class is `private`, `protected`, `public`, or inaccessible:

```
a          //inaccessible
b          //protected
c          //protected
setA       //protected
setB       //public
setC       //public
```

D) Suppose the `Quiz` class, derived from the `CheckPoint` class, is declared as class `Quiz : CheckPoint`
Is the `CheckPoint` class a `private`, `public`, or `protected` base class?
private

13. What will the following program display?

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

class First
{
	protected:
		int a;
	public:
		First( int x = 1)
			{ a = x; }

		virtual void twist()
			{  a *= 2;}

		int getVal()
			{ twist(); return a; }
};

class Second : public First
{
	private:
		int b;
	public:
		Second(int y = 5)
			{ b = y; }
		virtual void twist()
			{ b *= 10; }
};

int main()
{
	First object1;
	Second object2;

	cout << object1.getVal() << endl;
	cout << object2.getVal() << endl;
	return 0;
}

//Since class Second inherits from class First publicly
//therefore getVal is also defined for the Second class
//but the key thing to note is that the method twist is
//overridden in the class Second, therefore, when a Second
//object calls the getVal() method, the twist function associated
//with the Second class gets called, and not that of the First class
//and here arises the reason why the output is 2 1, since the
//overridden twist modifies b but not a, and therefore the default
//value of a is returned (it will never change with each successive
//getVal call, only b will change!)
//Output:
2
1
```

14. Which code fragment(s) could be inserted in the blank in order to safely initialize each element of A to zero?

```
int* p = &A[0]; //p = A
for (int Idx = 0; Idx < Size; Idx++, p++){
```

```
    _____;
}
```

1) `*A = 0;` 2) `A[Idx] = 0;` 3) `*p = 0;` 4) `*Idx = 0;` 5) All of the above
6) 1 and 2 only 7) 1 and 3 only **8) 2 and 3 only** 9) 1 and 4 only 10) None of these

15. _____ functions are dynamically bound by the compiler.
    A) **Virtual**
    B) Constructor
    C) Destructor
    D) Static
    E) None of the above
16. Class templates allow you to create one general version of a class without having to
    A) use private members.
    B) use member functions.
    C) **duplicate code to handle multiple data types.**
    D) write any code.
    E) None of the above

17. Polymorphism is when _____ in a class hierarchy perform differently, depending upon the class of the object making the call.
    A) **member functions**
    B) derived class constructors
    C) base class constructors
    D) derived class destructors
    E) None of the above

18. True/False: Pointers to a base class may be assigned the address of a derived class object.

TRUE

19. Assume we have the following class definition from the `IntVector` class.

```
class IntVector{
    private:
        unsigned sz;
        unsigned cap;
        int *data;
    public:
        IntVector();
        IntVector( unsigned size );
        IntVector( unsigned size, int value );
        ~IntVector();
       void doublesArray();

};
```

20. Write a definition for a function called `appendDouble`, that appends twice the value of the contents of the array. Ex. An array containing 2, 16, -12, 59 would contain 2, 16, -12, 59, 4, 32, -24, 128

```
void IntVector::appendDouble(void)
{
        unsigned oldSize;
        int *tempArray;

        if (sz*2 > cap)
        {
        oldSize = sz;
        sz *= 2;
        tempArray = new int[sz];
        for (unsigned i = 0; i < oldSize; i++)
        tempArray[i] = data[i];
        for (unsigned i = oldSize, k = 0; i < sz; i++, k++)
        tempArray[i] = data[k]*2;

        delete [] data;
        data = tempArray;
        }
        else
        {
        for (unsigned i = sz, k = 0; i < sz*2; i++, k++)
        data[i] = data[k]*2;
        sz *= 2;
        }
}
```