UPE Review

ADT's

- a coordinated group of data structures, algorithms, and interface functions
- data structures and algorithms
    - kept hidden from user
- interface functions
    - allow user to interact with data structures, use the algorithms

Constructors

- same name as the class | no return type | default constructor provided if and only if no user defined constructor exists

Initializer Lists

- another way to initialize member variables in the constructor
- useful in class composition
- `Car (int gallons) : m_numGallons(gallons);`

Construction/Destruction Order

- construction: build inner to outer (private first)
- destruction: opposite of construction, start with outer class

Copy Constructors

```
Engine e(8);
Engine f = e;    // copy constructor
Engine g(e);     // copy constructor
```

- Default copy constructor performs a **shallow copy**
    - all member variables copied over exactly
    - breaks down when variables are dynamically allocated pointers (**both objects share allocated memory!**)

Assignment Operator

```
Egine e(8);
Engine f(4);
f = e;
```

- default assignment operator performs **shallow copy**
    - same problem as with copy constructor

Linked Lists

- data structure containing:
    - a value
    - a pointer to the next item in the list
- normally made using C++ struct
- nodes contain value and pointers to next or previous node

```
assignmentOperator(ClassName& source) {
    Map temp(source);   // creates a temp Map using source
    temp.swap(*this);   // now temp has the Map *this did ("Hi")
    return *this;       // temp has "Hi" (which was dynamically allocated in insert())
}   // temp's destructor is called

int main() {
    ClassName a;
    ClassName b;
    a.insert("Hi"); // dynamically allocates using new (new Node)
    a = b;  // a is *this, b is source
}
```

Practice Problems

1.