CS 162 Recitation 5 Worksheet

1. Understand Program Errors (from Worksheet 4)
   a. Compiling Errors - The compiler GIVES you a line number and error message.
      E.g., syntax, missing a semicolon, use a variable before declaring it, etc.
      As a group/class, share your debug strategy:

   b. Runtime Errors - produce the wrong output or crash, leaving no clues as to why.
      E.g., Segmentation Fault
      As a group/class, share your debug strategy:

2. Pointer with structs:
   ```
   struct Pokemon {
      string name;
      int size_move;
      string *moves;
   }

   struct Pokedex {
      int num;
      Pokemon* p_arr;
   }

   void test( ___①___ var){
      // code
   }

   int main () {
      Pokedex p;
      p.num = 5;
      p.p_arr = new ___④___ [p.num];
      for (int i = 0; i < p.num; i++) {
         ...
         p.p_arr[i]_③_moves = new ___⑤___ [p.p_arr[i].size_move];
      }
      test (___②___);
      return 0;
   }
   ```
a. From line 17, is p a data type or an object?

b. What should be filled in ③ at line 22, a . or ->? What's the difference? Can we replace it with:
   p.p_arr->moves = ...
   Why or why not?

c. What should be filled in ④ at line 19?

d. What should be filled in ⑤ at line 22?

e. If ② is p, what should be filled in ①?

f. If ② is p.p_arr, what should be filled in ①?

g. If ② is p.p_arr[0], what should be filled in ①?

h. If ② is p.p_arr[0].moves, what should be filled in ①?

i. If ② is p.p_arr[0].moves[1], what should be filled in ①?

3. Accessor and Mutators:
   Create a garage class that has a dynamic array of vehicle structs. Make sure you create an int variable to indicate the number of vehicles and follow the rules for encapsulation. Write the declarations for mutator, and accessor functions needed to access the members in the garage. Use const when necessary.

   struct vehicle {
       string name;
       int num_wheels, num_seats;
       bool motor;
   };

   class garage {
       private:

           __________________________;    //dynamic array of vehicles

           __________________________;    //number of vehicles
public:

_____________________________; //accessor for dynamic array
_____________________________; //mutator for dynamic array
_____________________________; //accessor for number of vehicles
_____________________________; //mutator for number of vehicles

};

4. Use of const:
Given the following class declaration, explain each use of const. For a-d, tell what is legal, what is not illegal, and why.

```cpp
class MyClass {
private:
    int member1;
public:
    void fun1(const int x);
    int fun2() const;
};
```

a. void MyClass::fun1 (const int x){
    int y = x;
}

b. void MyClass::fun1 (const int x){
    x = member1;
}

c. int MyClass::fun2() const{
    return this->member1;
}

d. int MyClass::fun2() const{
    this->member1 = 2;
    return this->member1;
}