Programming Assignment 3
(Classes, Inheritance & Pointers)

Instructions:

- This assignment is to be turned in on Blackboard . You should submit '.cpp' files with names mentioned after each problem.

- Make sure your programs compile (without any compiler errors). You will not receive any credit if your program does not compile. If you are unable to complete any of the programs, submit the parts that work (with no compiler errors) for partial credit.

- Your grade will be based on functionality (does the program do what it is suppose to do), readability (is the code nicely formatted), and understandability (are the literals meaningful and is the code modular and well documented with appropriate comments). Try to incorporate all the good programming practices and styles. For example, **WRITE COMMENTS**, declare all constants and variables at the top of a function, store quantities using the appropriate data types, declare function prototypes, etc.

1. Inheritance and Polymorphism (30 points)

Design a *Ship* class that has the following members:
- A member variable for the name of the ship (a string)
- A member variable for the year that the ship was built (an int)
- Appropriate constructors, accessors and mutators
- A *print* function that displays the ship's name and the year it was built.

Design a *CruiseShip* class that is derived from the *Ship* class. The *CruiseShip* class should have the following members:
- A member variable for the maximum number of passengers (an int)
- Appropriate constructors, accessors and mutators
- A *print* function that overrides the *print* function in the base class. The *CruiseShip* class's print function should display only the ship's name and the maximum number of passengers.

Design a *CargoShip* class that is derived from the *Ship* class. The *CargoShip* class should have the following members:

- A member variable for the cargo capacity in tonnage (an int)
- Appropriate constructors, accessors and mutators
- A *print* function that overrides the *print* function in the base class. The *CargoShip* class's print function should display only the ship's name and the ship's cargo capacity.

To demonstrate the working of the classes create two objects of each type. The first object (for each type) must be created using the constructor that takes values for all of the data members. The second object must be created using the default constructor and mutators methods must be used to set the values of the data members. Also, make sure to get inputs from the user for each of the data member and NOT hardcode them in the program. You can define variables for getting these user inputs and then pass these variables to the constructors and the mutators. The program then should call each object's *print* function to display the data stored in them.

Filename: ship.cpp

---

2. Reverse (20 points)

Write a program that takes a string as input (from the user) and then reverses the contents of the string. Your program should work by using two pointers. The "head" pointer should be set to the address of the first character in the string, and the "tail" pointer should be set to the address of the last character in the string (i.e. the character before the terminating null). The program should swap the characters referenced by these pointers, increment "head" to point to the next character, decrement "tail" to point to the second to last character, and so on, until all characters have been swapped and the entire string reversed.

Hint: Use the length of the string to calculate the address of the last character in the string. Loop until head >= tail. You may have to convert your string variable to an array of characters.

Filename: reverse.cpp