Special Member Functions in C++11 Solutions

Move operators

- Modern C++ has two special member functions which are not present in traditional C++
- Describe them briefly
  - Move constructor - optimized version of copy constructor which moves its argument instead of copying it
  - Move assignment operator - optimized version of assignment operator which moves its argument instead of copy assigning it

Synthesized move operators

- In what circumstances will the compiler synthesize the move operators?
  - The compiler will synthesize move operators if the class does not define a copy constructor, assignment operator or destructor and every data member of the class is one of:
    - Built-in type
    - User-defined type with move operators
    - Static data member (not moved)
- How are these synthesized move operators implemented?
  - The synthesized move constructor will call the move constructor for each member
  - The synthesized move assignment operator will call the move assignment operator for each member

Synthesized copy operators in Modern C++

- If a class implements move operators but not copy operators, will the compiler synthesize the copy operators?
  - If a class defines a move constructor or a move assignment operator, the copy constructor and assignment operator will be synthesized as "deleted"
  - By default, the class will be move-only
  - The class should define its own copy operators if it needs them

Rule of Five

- What is the "Rule of Five"?
  - In Modern C++, the old "Rule of Three" has become the "Rule of Five" to accommodate move operators
  - If a class needs to implement a destructor to function correctly, then it probably needs to implement the copy and move operators as well

Move-only class

- Implement a simple move-only class
- Write a program to test your class. Verify that it cannot be copied