C++ Primer Plus, 5^{th} *Edition* by Stephen Prata
Chapter 11: Working with Classes
Review Questions

1. Using a member function to overload the multiplication operator for the `Stonewt` class; have the operator multiply the data members by a type `double` value. Note that this will require carryover for the stone-pound representation. That is, twice 10 stone 8 pounds is 21 stone 2 pounds.

   See the following code:

```
// prototype
Stonewt operator*(double x) const;

// definition
Stonewt Stonewt::operator*(double x) const
{
    double total_lbs = Lbs_per_stn*stn + lbs;
    return Stonewt(x * total_lbs);
}
```

2. What are the differences between a friend function and a member function?

   When overloading operators, specifically a binary operator, using a member function requires that the first operand be the object which invokes the function. If you use a friend, however, you may overload the operator to accept something other than the invoking object as the first operand with the invoking object as the second.

3. Does a nonmember function have to be a friend function to access a class's members?

   If the nonmember function is to access the class's members directly, then yes, it must be a friend function. If the nonmember function is to access the class's members indirectly (by invoking a member function, for example) then no, it does not need to be a friend function.

4. Use a friend function to overload the multiplication operator for the `Stonewt` class; have the operator multiply the `double` value by the `Stone` value.

   See the following code:

```
// function prototype
friend Stonewt operator*(double x, const Stonewt & s);

// definition
Stonewt operator*(double x, const Stonewt & s)
{
    return s*x;
}
```

5. Which operators cannot be overloaded?

We may not overload the following operators: `sizeof`
`.`
`.*`
`::`
`?:`
`typeid`
`const_cast`
`dynamic_cast`
`reinterpret_cast`
`static_cast`.

6. What restriction applies to overloading the following operators? `=`, `()`, `[]`, and `->`.
   The answer

7. Define a conversion function for the `Vector` class that converts a `Vector` object to a type `double` value that represents the vector's magnitude.
   The answer.