OOP244SCC Quiz 4:

Print Full Name:   
Seneca Email ID: **@myseneca.ca** Student number:

1. When classifying operators, they fall in three categories. Name the three categories.
2. There are two ways to overload an operator, Name them.
3. Explain what is a unary operator.
4. Explain what is a binary operator.
5. Explain what is a ternary operator.
6. Explain the difference between a member and a helper operator.
7. What are the four parts of the signature of a member operator.
8. What is the general signature of a binary member operator?
9. What is the general signature of a unary pre-fix member operator?
10. What is the general signature of a post-fix member operator?

Having the following class answer the upcoming questions:  
class Container {

int m\_volume;

int m\_capacity;

public:

Container();

Container(int volume, int capacity);  
// 11

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator\_\_\_\_\_();  
// 12

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator\_\_\_\_\_(\_\_\_\_);

// 13

­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator\_\_\_\_\_(\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_);

// 14

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_() \_\_\_\_\_\_\_;

// 16

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ operator\_\_\_\_\_  
  
 (\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ left, \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ right);

};

Fill in the blank in the questions below and their declaration above:

1. Overload the pre-fix “++” operator to add one to the volume of the Container only if the volume is less than the capacity. This operator returns the reference of the current Container object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_::operator\_\_\_\_\_\_\_\_\_\_\_(){

if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

return \_\_\_\_\_\_\_\_\_\_\_\_\_;

}

1. Overload the post-fix “++” operator to add one to the volume of the Container only if the volume is less than the capacity. This operator returns a copy of the current Container object before the addition.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_::operator\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ temp = \_\_\_\_\_\_\_\_\_\_\_\_\_;

if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

return temp;

}

1. Overload the “+=” operator to add an integer value to the volume of the Container only if the sum is not greater than the capacity and then return the volume

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_::operator\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_) {

if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

return m\_volume;

}

1. Overload the type conversion operator in a Container, to convert it to an integer by returning the volume.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_::\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_()\_\_\_\_\_\_\_\_\_ {

return \_\_\_\_\_\_\_\_\_\_\_\_;

}

1. Overload the “+” helper operator by reusing the += member operator. This operator returns a Container that has the sum of the volume of the left Container reference and the right Container reference operands.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator\_\_\_\_\_\_\_\_\_\_\_\_  
  
 (\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ left, \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ right) {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ temp = left;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

return temp;

}

1. Overload the “>” friend helper operator that returns true if the volume of the left Container operand is greater that the right Container operand.

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ operator\_\_\_\_\_  
   
 (\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ left, \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ right){

return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ > \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}