SCSSE, University of Wollongong
CSCI204/MCS9204 Object and Generic Programming in C++
Laboratory Week 6

Task One: Debugging

RectangleBlock.cpp and **RoomService.cpp** contain various errors; syntax, logical and/or with respect to the sample output. Copy the files to your own working directory and fix all bugs so the programs can be compiled and run correctly.

The correct output for RectangleBlock.cpp should be:

```
Rectangle...
Dimensions are 4 by 5
Area is 20
Block...
Height 6
Dimensions are 4 by 5
Volume is 120
```

The correct output for RoomService.cpp should be:

```
Room service order:
steak dinner $19.99
room service service fee $4 to room #1202
Total is $23.99
```

Task two: Overloading

Define a class Set in a file **Set.h** with private data member **elements (an integer pointer)** and **size** (integer) of the set. Define suitable member functions and overload binary operators +, -, and &. Define an overloading input operator (>>) to get elements from keyboard for a Set object, define an overloading output operator (<<) to print elements of a Set object. Don't forget to define a destructor.

We define the operator + for two sets as UNION (The set of UNION has all (non-repeated) elements in both sets);
We define the operator – for two sets as MINUS (The set of MINUS has all the elements in the first set but not in the second set);
We define the operator & for two sets as INTERSECTION (The set of INTERSECTION has only the elements in the first set and also belongs to the second set).

See the examples for more details.

Implement Set member functions and input / output operators in a file **Set.cpp**.

Write a main() function in **testSet.cpp** to test the overloaded operators (+, -, &, >> and <<).

When we run the program, the results may look like the following (red data mean inputs from keyboard):

```
Input number of elements for a set: 5
Input elements: 2 10 8 7 21
Input number of elements for a set: 3
Input elements: 6 3 10
Set s1 = {2, 10, 8, 7, 21}
Set s2 = {6, 3, 10}
UNION: s3 = s1 + s2 = {2, 10, 8, 7, 21, 6, 3}
MINUS: s3 = s1 - s2 = {2, 8, 7, 21}
INTERSECTION: s3 = s1 & s2 = {10}
```