CMPT 135: Lab Work Week 11

1. What is the output of the following program

```
template<class T1, class T2>
T1 sum_up(T1 a, T2 b)
{
	T1 result;
	result  = a + static_cast<T1>(b);
	return result;
}
```

```
int main()
{
		int a = 3;
		double b = 2.7;
		double answer;
		answer = sum_up(a, b);
		cout<< answer <<endl;
		answer = sum_up(b, a);
		cout<< answer <<endl;
		return 0;
}
```

2. Write a function template named maximum. The function takes two values of the same type as its arguments and returns the larger of the two arguments (or either value if they are equal).

3. Define a function template named `absValue` that taken one argument and returns the absolute value of the argument. Give some data types for the argument with which the function can be called.

4. Define a function template for binary search algorithm that can be used to search an array of elements of any type.

5. Define a function template for bubble sort algorithm that can be used to sort an array of elements of any type. Also define two more functions for selection sort and insertion sort algorithms.

6. Define a function template named `isIncreasing` that tests if the elements of an array are in increasing order.

7. Define a function template named `countElements` that returns how many times a given value is found in a given array.

8. Given the SmarterArray class template, add a member function named **`insert`** that takes an index and a value arguments so that the function first asserts the index argument is such that **`index >= 0 &&index <= size`** and then inserts the value at the required index in the calling object. If the index argument is equal to size, then this function should append the value to the calling object.

9. Given the SmarterArray class template we have implemented during lecture, add an operator member function == that returns true if the right hand side operand has the same number of elements as the left hand side operand and elements at corresponding indexes are identical. Otherwise returns false.

10. Given the SmarterArray class template we have implemented during lecture, add an operator member function + that returns a new SmarterArray object that is the concatenation of the operands.

11. Given the Map class template, add a member function named **`insert`** that takes an index and a key-value pair arguments so that the function first asserts the index argument is such that **`index >= 0 && index <= size`** and then inserts the key-value pair at the required index in the calling object. If the index argument is equal to size, then this function should append the key-value pair to the calling object.