CSCI204/MCS9204/CSCI804
Object and Generic Programming in C++
Laboratory Exercise 9 (Week 11)

Task 1: I/O manipulator (0.4)

Define a manipulator *format* in a file **task1.cpp**. The manipulator *format* can take three arguments: string of base (such as "oct", "dec" and "hex"), the length that an integer can be displayed, and a character can be used to pad the output value when the length of the value is not long enough.
Write a driver program include main() function in the file **task1.cpp** to test the manipular.
For example:

```
cout << format("oct", 10, '#') << 20 << endl;
```

The result is

```
########24
```

Change the format like

```
cout << format("hex", 8, '!') << 20 << endl;
```

The result is

```
!!!!!!14
```

Change the format like

```
cout << format("dec", 12, '@') << 20 << endl;
```

The result is

```
@@@@@@@@@@20
```

Task 2: String stream (0.6)

Use **ostringstream** and **istringstream** to implement the following ***two functions*** in a file **stringIO.cpp** by assuming that the operators << and >> have been overloaded for type T.

```
template<class T>
string toString (T value); // convert a value into a string

template <class T>
T toValue(string str); // extract the value from a string
```

Defined a class **Student** in a file **Student.h** and implemented the member functions in a file **Student.cpp**. The class Student can be defined like following

```
class Student {
        private:
                string firstname;
                string lastname;
                int id; // student number
                float gpa;
        public:
                // all necessary functions to be defined here
                … …
};
```

Implement a main() function in **stringIO.cpp** to test the two functions in the cases where T is **integer**, **double** and **class Student.**

You need to define and implement the student class properly in order to test the two template functions and assume the input string for the student class is in the following format.

First-name:last-name:student-number:gpa

For example:
John:Anderson:1234567:6.35

Testing:

Compile the program in this task by:
CC –o task2 stringIO.cpp Student.cpp

Run the program (You may use different values)
./task2
Input an integer: 12345678
Integer to string: 12345678
String to integer: 12345678
Input a double: 3214.654
Double to string: 3214.65
String to double: 3214.65
Input a student record (first-name:last-name:number:gpa): David:Smith:1234567:3.65
Student to string:
David:Smith:1234567:3.65
String to Student:
David:Smith:1234567:3.65

Note: The outputs above indicate different types of data.

Submission:

You should submit all the files to the banshee server by 11:59 PM on Friday, 16 October 2015 via command:

submit –u your-user-name –c CSCI204 –a L9 task1.cpp stringIO.cpp Student.h Student.cpp

and input your password.

Make sure that you use the correct file names. The UNIX system is case sensitive. You must submit all files in one *submit* command line.

After submit your assignment successfully, please check your email of confirmation. You should keep this email for the reference.

You would receive ZERO of the marks if your program codes could not be compiled correctly.

Later submission will not be accepted. Submission via e-mail is NOT acceptable.

End of Specification