Exercise 1:

Submit your solutions (source code) to the questions below by email to your instructor and your TA(s) by Tuesday, October 9th (16:30).

Question 1: A first C++ program (30 points).

Create a file named "hello.cpp". Complete the code below such that it prints out "hello!" on standard output. Use C++ streams.

```
// hello.cpp

// COMPLETE: import stream declarations

int main(void) {
  // COMPLETE: print out "hello!" on standard output using C++ streams
  return 0;
}
```

Compile the program by typing on your shell ("$>" is the shell prompt):

```
$> g++ hello.cpp -o hello
```

Test it by typing on your shell:

```
$> ./hello
```

Question 2: Reading from standard input (30 points).

Create a file "helloMyName.cpp". Write a program that reads your name from the standard input, then prints out on standard output "hello XX!", where XX is your name that was read from the standard input. Use C++ streams.

```
// helloMyName.cpp

// COMPLETE:
// import string declaration
// import stream declaration

int main(void) {
  std::string myName;
  // COMPLETE: read a string from standard input into the variable myName
  //   by using C++ streams
  // COMPLETE: print out "hello XX!" on standard output where XX corresponds
  //   to the string contained in the variable myName
  return 0;
}
```

Compile the program by typing on your shell ("$>" is the shell prompt):

```
$> g++ helloMyName.cpp -o helloMyName
```

Test it by typing on your shell:

```
$> ./helloMyName
```

Question 3: Converting numbers to strings (30 points).

Create three files: numToStringMain.cpp, numToString.cpp and numToString.h.

- numToString.h: declare three functions named intToString, floatToString and doubleToString that converts numbers to a string.
- numToString.cpp: implement the three functions above.
- numStringMain.cpp: defines a main function that tests the three functions above.

The prototype of the three functions is:

- string intToString(int)
- string floatToString(float)
- string doubleToString(double)

Each function takes respectively as argument an int, float and double and returns a string corresponding to the input number.

To implement these functions, use a stringstream object as a string buffer. Use the method str() of stringstream to generate a string.

The code for the main function is partially given below but needs to be completed:

```
// numToStringMain.cpp

// COMPLETE: include the proper headers

int main(void) {
  int numInt = 1;
  std::string str;
  // COMPLETE convert numInt to a string with the function intToString()
  //  and store the results in str
  // COMPLETE write str to the standard output

  float numFlt = 2.0f;
  // COMPLETE convert numFlt to a string with the function floatToString()
  //  and store the results in str
  // COMPLETE write str to the standard output

  double numDbl = 3.14;
  // COMPLETE convert numDbl to a string with the function doubleToString()
  //  and store the results in str
  // COMPLETE write str to the standard output

  return 0;
}
```

The code for intToString, floatToString and doubleToString is repetitive. At the end of this course, you will learn how to concisely write these functions by using template parameters.

Question 4: Check if an int contains letter in hexadecimal notation (10 points).

Write a function hexContainLetters(int) that returns true if the integer passed as argument contains letter in its hexadecimal representation. In order to do that the following classes and methods will be needed:

- stringstream
- the method .fail() of the class stringstream that returns true if the stream is in an error state
- the stream modificators hex and dec

Define this function in a file named hexContainLetters.cpp.

To test your function, you can use the following code (write it in a file named testHexContainLetters.cpp).

```
// testHexContainLetters.cpp

#include <cassert> // for assert()
// COMPLETE: import the proper header
```

```
int main(void) {
  int i = 1;
  assert(hexContainLetters(i) == false);

  i = 10;
  assert(hexContainLetters(i) == true);
  return 0;
}
```