Queens College, CUNY, Department of Computer Science
Object-Oriented Programming in C++
CSCI 211/611
Summer 2018
Instructor: Dr. Sateesh Mane

due date Tuesday, July 24, 2018, 11.59 pm

Homework: namespaces

- **Experience with other classes has demonstrated that in many cases the source of difficulty is not the mathematics or the programming.**
- **The source of difficulty is the English (understanding the text).**
- **If you do not understand the words in the lectures or homework, THEN ASK.**
- **If you do not understand the concepts in the lectures or homework, THEN ASK.**
- **Send me an email, explain what you do not understand.**
- **Do not just keep quiet and then produce nonsense in exams.**

- **Consult your lab instructor for assistance.**
- **You may also contact me directly, but I cannot promise a prompt response.**
- Please submit your inquiry via email, as a file attachment, to `Sateesh.Mane@qc.cuny.edu`.
- Please submit one zip archive with all your files in it.
  1. The zip archive should have either of the names (CS211 or CS611):

     ```
     StudentId_first_last_CS211_hw_namespaces.zip
     StudentId_first_last_CS611_hw_namespaces.zip
     ```

  2. The archive should contain one "text file" named "hw_namespaces.[txt/docx/pdf]" (if necessary) and cpp files named "Q1.cpp" and "Q2.cpp" etc.
  3. Note that a text file is not always required for every homework assignment.
  4. Note that not all questions may require a cpp file.

General information

- You should include the following header files, to run the programs below, unless otherwise stated in the question.

```
#include <iostream>
#include <iomanip>
#include <string>
#include <vector>
#include <cmath>
```

- If you require additional header files to do your work, feel free to include them.
- **Include the list of all header files you use, in your solution for each question.**
- The questions below do not require complicated mathematical calculations.
- If for any reason you require help with mathematical calculations, **ask the lab instructor or the lecturer.**

Q1 Do *not* use namespace std

- **Do *not* use the namespace std.**
- Edit the code in the main function below to make it run correctly.

```
#include <iostream>
#include <string>
#include <cmath>

// DO NOT WRITE "using namespace std"

int main()
{
  string s("not using std");                    // edit

  cout << s << endl;                            // edit

  double x = -1.234;
  cout << "abs(x) = " << abs(x) << endl;        // edit

  return 0;
}
```

Q2 Namespaces Student1,2,3

- You are permitted to use the namespace std in this question.
- Here are three functions written by three students.
  1. The first two functions are slightly peculiar, for different reasons.
  2. *Do not edit the functions. They are actual student code, submitted as homework solutions.*

```
void sum_array(int j, const double *a, double &sum)
{
  for (int i = 0 ; i < j ; ++i)
    sum = sum + a[0];
}

void sum_array(int j, const double *a, double &sum)
{
  sum = 0;
  for (int i = 0; i <= j; i++) {
    sum += a[i];
  }
}

void sum_array(int j, const double *a, double &sum)
{
    sum = 0;
    for (int i = 0; i < j; i++)
    {
        sum += a[i];
    }
}
```

- **Place the functions in three namespaces "`Student1, Student2, Student3`" and make the fillowing main program run correctly.**

```
#include <iostream>

using namespace std;

// namespaces and functions


int main()
{
  int n = 3;
  double a[] = {1.1, 2.2, 3.3};
  double sum1 = -1.0;
  double sum2 = -2.0;
  double sum3 = -3.0;
  Student1::sum_array(n, a, sum1);
  Student2::sum_array(n, a, sum2);
  Student3::sum_array(n, a, sum3);

  cout << "Student1: " << sum1 << endl;
  cout << "Student2: " << sum2 << endl;
  cout << "Student3: " << sum3 << endl;

  return 0;
}
```

Q3 Namespaces N1, N2

- You are permitted to use the namespace std in this question.
- **Place the following code in a namespace N1.**

```
void print() {
  cout << "print N1" << endl;
}

class Stuff {
public:
  Stuff(int n, double x);
  int get_i() const;
  double get_d() const;
private:
  int i;
  double d;
};
```

- **Place the following code in a namespace N2.**

```
void print() {
  cout << "print N2" << endl;
}

class Stuff {
public:
  Stuff(string s);
  string get_str() const;
private:
  string str;
};
```

- **Write the non-inline function definitions and place them in different blocks of the namespaces N1 and N2.**

```
namespace N1 {
// non-inline function definitions for class Stuff of namespace N1
}

namespace N2 {
// non-inline function definitions for class Stuff of namespace N2
}
```

- **Use your code to make the following main program run correctly.**

```
#include <iostream>
#include <string>

using namespace std;

// namespaces N1 and N2

int main()
{
  int n = 4;
  double x = 2.345;
  string s("abcd");
  N1::Stuff a(n, x);
  N2::Stuff b(s);

  cout << "N1: " << a.get_i() << "  " << a.get_d() << endl;
  cout << "N2: " << b.get_str() << endl;

  N1::print();
  N2::print();

  return 0;
}
```