**C++11/14 Exercises**

**Student: Javier Montes Hermosilla**

**Exercises**

C++11 Online compiler (All exercises can be made here): <https://www.onlinegdb.com/online_c++_compiler>

1. **Key Questions**

**What is the difference between NULL and nullptr?**

**Nullptr is implicitly convertible and comparable to any pointer type.**

**But unlike NULL, it is not implicitly convertible or comparable tointegral types.**

**When should we use Lambda Functions?**

**When we want to**

**demonstrate some kind of code that needs to be run, but we don’t**

**want to make a FULL function and we would rather use them more**

**as a variable**

**Can strongly-typed enums be compared if they represent a different class?**

**No.**

**The use of word class implies that each enum**

**type really is different and it can’t be compared with other enum types.**

**For what purpose is Noexcept used?**

**When you are building a**

**function, method, or lambda, this part of the code won’t throw any exception, and it is used to check, during compilation time, if an**

**expression throws or not an exception.**

**What’s the main goal of Static Assert?**

**To use it to check if a condition it’s TRUE when the code is already compiled,**

**but if it isn’t, then the compiler will show an error message and will stop**

**compiling.**

1. **Refactor a given program code into C++11 code using lambda expressions and typedef for the integers.**

#include <iostream>

#include <algorithm>

#include <vector>

#include <cstdlib>

using namespace std;

void randomassign(int& pos)

{

int r = (rand() % 100) + 1;

pos = r;

}

void print(int pos)

{

cout << pos << endl;

}

int main()

{

vector<int> vector(15);

for\_each(vector.begin(), vector.end(), randomassign);

for\_each(vector.begin(), vector.end(), print);

return 0;

}

1. **Code a simple program of your choice into C++11 using at least 5 of the new available features previously explained. I recommend implementing Lambda Function and Static Assert for the sake of more interesting problem to be solved.**

In this exercise students are required to make a new C++ program of their choice that uses at least 5 of the 10 mentioned new features. I suggest students to try and implement every feature separately, so that they can see and experience by their own the workings of this new features I have shown in lecture, and then choose their favorites into this simple program.

Remember the features we’ve seen are:

* Type inference (Auto, decltype)
* Range-based for
* Nullptr
* Constexpr/const
* Typedef
* Noexcept
* Lambdas expressions
* C++11 new initialization
* Strongly-typed enums
* Static assert
* Explicit Overrides
* Multithreading