Universidad Zaragoza

1542

Graphics and Computer Vision

MEMORY MANAGEMENT AND AN INTRODUCTION TO POINTERS

Fundamentals on Computing for Robotics,

Prerequisites

In this laboratory, you are going to write several small programs covering the basic aspects of pointers. Please use the same compiler and setup that in previous laboratories. Please do not forget to use these CXXFLAGS when compiling with g++: `–Wall –Wextra –pedantic –Werror`. These flags would help you to write more solid code.

Do not hesitate to contact the faculty in case of doubts about the exercises.

Identifying variables and pointers

Please write a program (named `global_local.cpp`) that has a global integer variable storing the value 1, a local integer variable (of the main function) storing the value 2. Also, the program should have two pointer to integer variables pointing to the local and the global variables. Finally, the program will print the values of the four variables with the following format:

```
global_var stores 1
local_var stores 2
global_ptr points to 0x<correct value>
local_prt points to 0x<correct value>
```

In which section each variable will be stored?

Modifying variables with pointers and references

Please write a program (named `write_through_ptr_ref.cpp`) that: 1) reads an integer number from the standard input using std::cint, 2) increments the value by 2 with a constant pointer, 3) prints he updated value in the screen, 4) substracts the value by 2 with a reference, and 5) prints the final value.

Fix the code

Double delete

Could you please fix the following program named `to_fix_double_delete.cpp`. Please run the original program before writing the fix.

```
int* create_int_ptr()
{
   int* ptr = new int;
   return ptr;
}

void delete_int_ptr(int* ptr)
{
```

```
   delete ptr;
}

int main()
{
   int* int_ptr=create_int_ptr();
   delete_int_ptr(int_ptr);
   delete int_ptr;
}
```

Missing new

Could you please fix the following program named `to_fix_missing_new.cpp`. Please run the original program before writing the fix.

```
#include <iostream>

int* create_two_array()
{
   int* ptr = new int[2];
   return ptr;
}

void delete_two_array(int* ptr)
{
   delete ptr;
}

int main()
{
   int* array=create_two_array();

   array[0] = 6;
   array[1] = 7;

   std::cout << "array[0]: " << array[0] << std::endl;

   delete_two_array(array);
   array=reinterpret_cast<int*>(0xDEADBEEF); // Is 0xDEADBEEF a valid address?

   std::cout << "array[1]: " << array[1] << std::endl;
}
```

Avoid using pointers when possible

Please read the `variables.cpp` program and briefly describe what the program does, specially in terms of pointer arithmetic.

```
#include <iostream>

namespace { // anonymous namespace

const size_t n = 6;
unsigned int lynn_conway[n] = {3, 4, 7, 8, 12, 2};

bool is_even(unsigned int a) {
```

```
  unsigned int mask = 0x1;
  return (a & mask) == 0;
}

} // anonymous namespace

int main() {

  unsigned int *lynn_ptr=lynn_conway;

  for(size_t i = 0; i < n; ++i) {
    if(is_even(lynn_ptr[i])) {
      std::cout << *(lynn_ptr+i) << " is even." << std::endl;
    }
  }
}
```

Then, what would happen if n is changed to 7 in the loop. ¿Does the compiler gives any error?

And, finally, could you remove the pointers and use a range-for loop. The final file should be named `variables_wout_pointers.cpp`.

Pointer arithmetic

Please write a program that finds the maximum and the minimum elements in a four by four float 2D matrix. The first version, `max_min_global_2d_matrix.cpp` of the program should store the matrix in the data section, and should access the elements with a pointer to float. The second version, `max_min_heap_2d_matrix.cpp`, should read two integer numbers from the command line and create a matrix in the heap with those dimensions, always less than 5 rows and columns. To fill the matrix, use the C++ standard library pseudo-random number generators ; e.g.:

```
#include <random>

std::random_device rd;
std::mt19937 gen(rd());
std::uniform_real_distribution<float> dis {0.0f, 1.0f};

auto pseudo_random_float_value = dis(gen);
```

The output of both programs should be:

```
The maximum and minimun values of the matrix are: XXX and YYY
```

Notes: Could you ask yourself how the memory stores the matrix? Do you need to delete the matrix in any case?

Submission

Please group the all the files within a zip. In those assignments with text questions, add then at the beginning of the file as C++ comments.

Before submitting, remember to talk to your teacher and have a **small interview** describing and answering questions about the job you have done and the decissions you have made. This interview is *20%* of the total grade of the assignment, while the submitted material is evaluated for the other *80%*.

All source code files, must include a comment on top with the names and NIAs of the students involved on their development. Then, all source code files, including directory structure, must be compressed into a single `zip` file with the following naming:

- **`memory_management_<nia1>_<nia2>.zip`**, where `<nia1>` and `<nia2>` are the NIAs of the involved students, if the work has been done in pairs. In this case, **only one** of the students must submit the work in Moodle.
- **`memory_management_<nia>.zip`**, where `<nia>` is the NIA of the involved student if the work has been done individually.

The compressed file must not contain anything else besides the source code files. Particularly, do not submit any binary file, neither executable, library nor object. The submission of the `zip` file must be done through the corresponding task in Moodle before the stablished deadline.