# CS 280 - Programming Assignment Object Allocator

**Information**

* [Handout](docs/Handout.pdf)
* Interface: [Text](code/ObjectAllocator.h)
* Sample driver program: [Text](code/driver-sample.cpp)
* Random number generator
  + Interface: [Text](code/PRNG.cpp)
  + Implementation: [Text](code/PRNG.cpp)
* Sample command lines for compiling:
  + Microsoft:

**cl** -Fems.exe driver-sample.cpp ObjectAllocator.cpp PRNG.cpp /EHa /MTd /W4 /WX /Za /D\_CRT\_SECURE\_NO\_DEPRECATE

* + Borland (Codegear/Embarcadero):

**bcc32** -w **-vG -v -w-8026** -ebor.exe driver-sample.cpp ObjectAllocator.cpp PRNG.cpp

GNU:

g++ -o gnu.exe driver-sample.cpp ObjectAllocator.cpp PRNG.cpp -Wall -Wextra -Wconversion -pedantic -Wl,--enable-auto-import

* + Running under Valgrind in Linux:

**valgrind** -q --leak-check=full --show-reachable=yes --tool=memcheck ./gnu.exe 1

* Output:
  + 32-bit: [With addresses](data/sample_outputs/output-sample-addresses-ILP32.txt)    [Generic (no addresses)](data/sample_outputs/output-sample-zeros-ILP32.txt)  **Updated 5:50 pm 17/9/2013**
  + 64-bit (LLP64): [With addresses](data/sample_outputs/output-sample-addresses-LLP64.txt)    [Generic (no addresses)](data/sample_outputs/output-sample-zeros-LLP64.txt)
  + 64-bit (LP64): [With addresses](data/sample_outputs/output-sample-addresses-LP64.txt)    [Generic (no addresses)](data/sample_outputs/output-sample-zeros-LP64.txt)
* Output with extra credit:
  + 32-bit: [With addresses](data/sample_outputs/output-sample-addresses-ec-ILP32.txt)    [Generic (no addresses)](data/sample_outputs/output-sample-zeros-ec-ILP32.txt)   **Updated 5:50 pm 17/9/2013**
  + 64-bit (LLP64): [With addresses](data/sample_outputs/output-sample-addresses-ec-LLP64.txt)    [Generic (no addresses)](data/sample_outputs/output-sample-zeros-ec-LLP64.txt)
  + 64-bit (LP64): [With addresses](data/sample_outputs/output-sample-addresses-ec-LP64.txt)    [Generic (no addresses)](data/sample_outputs/output-sample-zeros-ec-LP64.txt)
* [Diagrams](docs/example_in_diagrams/Diagrams.pdf) to help you understand the situation.
* [FAQ](docs/FAQ.pdf)for this assignment.
* [One Past the End](docs/One%20Past%20the%20End.pdf) This is a refresher on the C/C++ concept of pointing/dereferencing an element "one past the end" of an array. Going more than one past the end is **undefined**. Don't do it. You will lose points if you do. Plus, it's easy to detect using CodeGuard.

**Other Criteria**

* You must use **new** and **delete** to allocate/deallocate the pages.
  + Do not use **std::nothrow** with **new**. (If you don't know what that is, you're probably not going to use it.)
  + You must catch the exception (**std::bad\_alloc**) that is thrown from **new**.
* You can't use any STL code in your implementation (with the exception of **std::string** for exception messages).
* You can't change the **public** interface at all.
* Given this struct:

**struct** Student

{

**int** Age;

**float** GPA;

**long** Year;

**long** ID;

};

**sizeof**(Student) returns:

32-bit: 16

LLP64: 16

LP64: 24

* DO NOT assume that pointers, integers, and long integers are the same size. Doing so may cause the compiler to emit warnings and can also lead to your program crashing. See [this](docs/64.pdf)for a quick refresher.

**Exception Handling Examples**

Please note that these are *examples*! Your code will probably be different.

*// Make sure this object hasn't been freed yet*

**if** (is\_on\_freelist(anObject))

**throw** OAException(OAException::E\_MULTIPLE\_FREE, "FreeObject: Object has already been freed.");

*// If new throws an exception, catch it, and throw our own type of exception*

**char** \*newpage;

**try**

{

newpage = **new** **char**[PageSize\_];

}

**catch** (std::bad\_alloc &)

{

**throw** OAException(OAException::E\_NO\_MEMORY, "allocate\_new\_page: No system memory available.");

}

**if** ( */\* Object is on a valid page boundary \*/* )

*// put it on the free list*

**else**

**throw** OAException(OAException::E\_BAD\_BOUNDARY, "validate\_object: Object not on a boundary.");

[Download the executable](test/OATestGUI-pages.exe) This is an older version, but it includes the more sophisticated "hex viewer" component on the Page list tab.