**CS 4TB3 Final Project Proposal**

Group 01

**Project: Extending P0 with explicit Deallocation**

**Group Members:**

Ziqing Xu, 400079409

Yujing Chen, 001303292

Baikai Wang, 400084727

**Description:**

The task of the project is to extend P0 with heap-allocated objects that are explicitly deallocated when it is no longer needed. The developing challenge of this project is how to manage a linked-list of free blocks on the heap and how to test whether the heap-allocated objects are deallocated successfully or not. There are two ways for testing the implementation. First, dangling reference can be used to test whether a heap-allocated object is explicitly allocated or deallocated successfully.For example, X is a heap-allocated object and Y refers to X. We can test if X is allocated or deallocated by testing whether Y is valid or not. Secondly, we can test it by checking the symbol table seeing whether the deallocated heap-allocated object deleted successfully or not. For the documentation, this project will be documented by a README file, API documentation and code documentation. In the README file, it contains a brief description of the project, issues that detected during the implementation, and citation and licensing information. The API documentation will include what a function does and what a function returns. In addition, the source code will be documented with comments which are beneficial for others to easily read and re-use the code. In this project, we hope to develop a deeper understanding about P0 compiler structure and the implementation of heap allocation.

**Resources:**

#### Software:

#### Jupyter Notebook [Python]

* + Nbimpoter: <https://github.com/grst/nbimporter>
  + Webassembly: <https://webassembly.org/>

#### Articles:

#### <https://stackoverflow.com/questions/4495990/how-to-implement-a-memory-heap>

#### How to write a documentation <https://guides.lib.berkeley.edu/how-to-write-good-documentation>

#### Software:

* + <https://www2.cs.arizona.edu/~collberg/Teaching/553/2011/Handouts/Handout-8.pdf>
  + <https://avenue.cllmcmaster.ca/content/enforced/308459-COMPSCI_4TB3_emil_2201/8%20Further%20Data%20Types.pdf?_&d2lSessionVal=SBakUfjCaIiRo6aneEwcNJwlF>

#### Software:

* + <https://docs.python.org/3/c-api/allocation.html>

**Division of work**

|  |  |
| --- | --- |
| milestone | goal |
| API | Design API - what classes, functions should be implemented; which packages/tools are used to implement |
| Heap-objects Allocation | Handling the heap object allocation request, new(p). Heap manager will search for a free block on the heap which is big enough for the requested size. If there is no such block, the heap manager will ask for a big block of memory from the OS for handling the allocation request. |
| Heap-objects Explicitly Deallocation | Implement dispose(p) which could reclaim the memory occupied by p^. We use a list to keep track of available memory and dispose(p) would free blocked memory and adding its address and size back to the list. |
| Testing & Documentation | Testing the implementation and debugging; Providing documentation of the project |
| Poster & Submission | Design Poster; Submit all the source codes, pdf of the poster, tests and documentation. |

**Timeline:**

|  |  |
| --- | --- |
| milestone | Date |
| API | Feb 26 – Mar 3 |
| Heap-objects Allocation | Mar 4 – Mar 10 |
| Heap-objects Explicitly Deallocation | Mar 11 – Mar 17 |
| Testing & Documentation | Mar 17 – Mar 24 |
| Poster & Submission | Mar 25 – April 1 |