|  |  |
| --- | --- |
| Name |  |
| KUID |  |

# EECS 168 2020 Fall: Final Exam

Possible points: 130pts

Time Limit: 2.5 hours

Submit via email to your Lab TA

Place an X in the box that applies to you:

|  |  |
| --- | --- |
| I am in the MWF section (officially) |  |
| I am in the TR section (officially)  |  |
| I am an EECS 169 student (leave blank if not in 169) |  |

#

# Rules

#

**You will receive a zero for this exam unless you read the following and answer below**

Hi students,

I know this has been a tough, stressful, completely unprecedented semester. I really do appreciate all the effort you've put into this course.

Unfortunately, before we get started I need to remind everyone of the following rules:

* You must take this exam by yourself, no communicating with anyone else during the exam
* DO NOT use unauthorized aid such as chegg, google searches, exams from past semesters, coding forums, etc.
* DO NOT use a compiler

You 100% are allowed to look at:

* Any notes, code, or videos from the class wiki
* Your notes and code

In short, I need you to agree that you will take this exam by yourself and in an honest way. I need you to think of someone in your life, other than me, that would be sorely disappointed if they knew you cheated and write their name below.

*I agree to follow the rules for this exam because I want to do right by the following person…*

|  |  |
| --- | --- |
| Answer |  |

# [30pts] Conceptual

**Provide your answers in the given boxes.**

1. [3pts] When is a stack allocated variable that was declared in a function named *bananarama* deallocated?

|  |
| --- |
|  |

1. [3pts] When is a heap allocated array that was created in a function named *bananarama* deallocated?

|  |
| --- |
|  |

1. [3pts] How many times, in a single object’s life, is a constructor called?

|  |
| --- |
|  |

1. [3pts] When an object is deallocated what method is called?

|  |
| --- |
|  |

1. [3pts] What does the keyword *const* following a method definition do?

|  |
| --- |
|  |

1. [3pts] If a class has exactly one constructor and that constructor requires a parameter, can you make an array of objects of that type, yes or no?

|  |
| --- |
|  |

1. [3pts] What details of a stack allocated array must be decided on at declaration?

|  |
| --- |
|  |

1. [3pts] Why will the equation *area = (1/2)\*base\*height* always results in zero even if the base and height are greater than zero?

|  |
| --- |
|  |

1. [6pts, 2pts each] Assume a Circle class has a private member variable named m\_radius. Assume "Circle.h" is included as needed. Answer the following questions below indicating whether or not the attempted access is legal or illegal by placing an 'X' in the appropriate column.

|  |  |  |
| --- | --- | --- |
| **Code** | **Legal** | **Illegal** |
| int main(){ Circle c1; c1.m\_radius = 5.5.; //main continues... |  |  |
| int main(){ Circle\* c1 = new Circle(); c1->m\_radius = 5.5.; //main continues... |  |  |
| void changer(Circle c){ c.m\_radius = 5.5;} |  |  |

# [25pts] Code Snippets

**For these questions you'll only need to write enough code to answer a given question.**

[4pts] Write only the needed code to create a 2D heap allocated array of doubles with 3 rows and 10 columns. Initialize them all to 55.

|  |
| --- |
|  |

[2pts] Write the needed code to deallocate your 2D array of doubles. (Variable names must be consistent with the previous question!)

|  |
| --- |
|  |

[4pts] Assume there is a class called Mystery. It has a constructor that takes zero parameters. It has a public method with the signature:

 *bool setMember(int value)*

This method takes an int and returns a bool. It returns true if the call to setMember was successful and false otherwise. Let the user choose a value that you then pass to setMember. If the call was successful, print "finished" otherwise print "error."

|  |
| --- |
|  |

[15pts] Write the needed code to open a file named "final-exam.txt" which you can assume contains 77 doubles. Read in and print all 77 doubles to terminal. Finally, let the user choose a file name. You will then write the 77 doubles out to that file. Make sure not to cause any memory leaks.

|  |
| --- |
|  |

# [45pts] Classes: Memory Visualization

The following section will require you to look at and cross reference the code provided below and on the following page.

Below is the header file for the Course class we implemented in lecture. You may assume the methods listed are implemented properly.

|  |
| --- |
| #ifndef COURSE\_H#define COURSE\_H#include <string>class Course{ public: Course( std::string dept, int number ); ~Course();  std::string getDept() const; int getNumber() const;  //Define the overloading of == bool operator==(const Course& rhs) const;  //Copies all the ids into the course's array bool enroll(int\* roster, int numStudents);  private: std::string m\_dept; int m\_number; int\* m\_ids; //Array of ids int m\_numEnrolled; //Size of the array };#endif |

Trace the following code up to the lines labeled "FREEZE POINT 1" and then "FREEZE POINT 2" then answer the questions that follow regarding memory allocation.

|  |
| --- |
| int main(){ const int SIZE = 4; int nums[SIZE]; nums[0] = 759; nums[1] = 663; nums[2] = 222; nums[3] = 168; Course math125("MATH",125 ) ; Course\* math126 = nullptr; math125.enroll(nums, SIZE); math126 = new Course("MATH", 126); math126->enroll(nums, SIZE); //FREEZE POINT 1  Course\* math126b = nullptr; math126b = math126; delete math126b; //FREEZE POINT 2; return(0);} |

Trace the code up to FREEZE POINT 1 and answer the following questions. Please note, these questions are ALL OR NOTHING; no partial credit.

1. [3pts] How many Course objects are allocated?

|  |
| --- |
|  |

1. [3pts] How many arrays are allocated?

|  |
| --- |
|  |

1. [3pts] How many Course objects are on the call stack?

|  |
| --- |
|  |

1. [3pts] How many Course objects are on the heap?

|  |
| --- |
|  |

1. [3pts] How many arrays are on the calls stack?

|  |
| --- |
|  |

1. [3pts] How many arrays are on the heap?

|  |
| --- |
|  |

1. [3pts] Which ids (if any) are enrolled in math125. List them below.

|  |
| --- |
|  |

1. [3pts] Which ids (if any) are enrolled in math126. List them below.

|  |
| --- |
|  |

Trace the code up to FREEZE POINT 2 and answer the following questions

1. [3pts] How many Course objects are allocated?

|  |
| --- |
|  |

1. [3pts] How many arrays are allocated?

|  |
| --- |
|  |

1. [3pts] How many Course objects are on the call stack?

|  |
| --- |
|  |

1. [3pts] How many Course objects are on the heap?

|  |
| --- |
|  |

1. [3pts] How many arrays are on the calls stack?

|  |
| --- |
|  |

1. [3pts] How many arrays are on the heap?

|  |
| --- |
|  |

1. [3pts] Which ids (if any) are enrolled in math126b. List them below.

|  |
| --- |
|  |

# [30pts] Classes: Implementation

Below is the header file for the Account class we made in lecture. You may assume that the constructor, deposit, withdraw, checkBalance, and transfer methods are implemented properly. The other methods, you will need to implement in the questions that follow.

|  |
| --- |
| #ifndef ACCOUNT\_H#define ACCOUNT\_Hclass Account{ public: //Assume these are implemented Account(double startingBalance);  bool deposit(double amount); bool withdraw(double amount); double checkBalance() const; bool transfer(Account& destination, double amount);  //You will implement the following on the next pages bool operator<(const Account& rhs) const;  private: double balance;}; |

1. [5pts] In the box below, implement the operator> method which should return true if the Account on the left hand has more of a balance than the right hand side.

|  |
| --- |
| bool Account::operator>(const Account& rhs) const{ //your code below |

1. [25pts] Assume you are in main.cpp and the entire Account class is implemented correctly (even if you didn't know how to implement the > operator). In main, you must:
* Obtain a minimum balance from the command line
	+ example: $>./prog 55
	+ The command line above would set the minimum balance (used later in the program) to 55.
* Let the user open exactly 10 accounts and set the starting balance for each account (Accounts can have starting balance lower than the minimum balance if the user chooses)
* Print the a count of how many accounts have a higher balance than the minimum balance (set by the command line argument) to the terminal.
* Make sure to have zero memory leaks

You will write you answer on the following page

|  |
| --- |
| //main.cpp#include "Account.h"#include <string>int main(int argc, char\*\* argv){ //Your code below |

This should be page 16.

If it is not, please make sure you didn't alter the format of the exam.

The exam is over.

Email it to your TA.

Thank you for your hard work this semester!