CSE 333 Midterm Exam
April 27th, 2012

Your Name: _____ Solution Set _____

Student ID: _____

General Information:

This is a closed book examination. You have **50 minutes** to answer as many questions as possible. The numbers in parentheses at the beginning of each question indicate the number of points given to the question. There are **10 pages** on this exam (check to make sure you have all of them), and there is a total of **100 points** in all. Write all of your answers directly on this paper. Make your answers as concise as possible. If there is something in the question that you believe is open to interpretation, then please go ahead and interpret, but state your assumptions in your answer.

If you need to tear a page out of the midterm to avoid flipping back and forth, do so carefully; don't tear out any pages with your writing on it, and be careful of the staple!

	Points available	**You earned**
Problem 1	25	
Problem 2	30	
Problem 3	20	
Problem 4	25	
Problem 5	+1	
Total:	**100**	

Problem 1: Pointer precision, please (25 points)

What is the output of the following C program? (The lines that lead to the printing of output are in bold.). There is space for your answer on the next page.

```
#include <stdio.h>
#include <stdlib.h>

void displayarray(int *x, int len) {
  int i;
  for (i = 0; i < len; i++) {
    printf("%d", x[i]);
    printf("%s", (i == len-1 ) ? "\n" : " ");
  }
}

int main(int argc, char **argv) {
  int a1[] = {1, 2, 3, 4, 5, 6, 7};
  int a2[3] = {1, 3, 5};
  int i, *ptr;

  displayarray(a1, 7);

  for (i=0; i<3; i++) {
    int old;
    ptr = &(a1[0]) + a2[i];
    old = ptr[1];
    ptr[1] = ptr[0];
    ptr[0] = old;
  }
  displayarray(a1, 7);

  (*(ptr-1))++;
  displayarray(ptr-2, 2);

  return EXIT_SUCCESS;
}
```

1 2 3 4 5 6 7
1 3 2 5 4 7 6
5 5

(***more space for problem 1.)***

Problem 2: Bug bashing (30 points)

The following program has bugs. Circle the bugs, write code that repairs the bugs, and if your bugfix is on a different line than the bug, write a comment next to the bugfix with the bug line number that it fixes. Ignore stylistic issues; find and fix actual bugs. There are a minimum of 15 changes you need to make. (!!)

```
1   #include <iostream>
2   #include <stdlib.h>
3   #include <string>
4   using namespace std;
5   // A class that stores two different things.
6
7   template <class X, class Y> class Pair {
8
9    public:
10
11    Pair(X a, Y b) { first_ = a; second_ = b; }
12
13    void Set(X &a, Y &b) {  // &, not *
14      first_ = a;    // a, not *a
15      second_ = b;   // b, not &b
16    }
17
18    void Get(X *a, Y *b) {
19      *a = first_;
20      *b = second_;
21    }
22
23   private:
24    X first_;
25    Y second_;
26  };
27
28  int main(int argc, char **argv) {
29
30    Pair<string,int> si_pair("hello", 2);
31
32    Pair<int,string> *is_pair =
33        new Pair<int,string>(3, "there");
34
35    string a;
36    int b;
37
38    si_pair.Get(&a, &b);
39    is_pair->Set(b, a);  // b, a   not   *a, &b
40
41    delete is_pair;    // delete, not delete[]
42    return EXIT_SUCCESS;
43  }
```

(***more space for problem 2.)***

Problem 3: C Code Can Copy. (20 points)

In this problem, you will write some C code. Your job is to:

a) in file "copyarray.c", define a function CopyArray() that:

i. accepts an array of integers as a first parameter
ii. accepts an integer (the length of the array) as a second parameter
iii. has a third parameter that is an output parameter (see v.)
iv. allocates space for a copy of the array, and copies the array into it
v. returns the newly allocated/copied array through the output parameter (see iii.)
vi. returns "1" on success and "0" on failure

b) in file "copyarray.h", declare a prototype for CopyArray().

c) in file "copymain.c", implement main() to test the correctness of CopyArray().

Be sure to write defensive code. Also, make sure your code has no memory leaks. Feel free to use C standard library functions to get the job done, but #include the right header if you do so. If there are corner cases, do something reasonable to handle them. You have this page, plus two blank pages, to write your code.

copyarray.h:

```
/*
 * CSE333 Midterm Solution Set, Spring 2012
 *
 * Copyright 2011 Steve Gribble
 */

#ifndef _COPYARRAY_H_
#define _COPYARRAY_H_

// Allocates space for a new int array of length "arrlen", copies
// "arr" into it, and returns the new array through the output
// parameter "outputarr".

int CopyArray(int *arr, int arrlen, int **outputarr);

#endif // _COPYARRAY_H_
```

(*more space for problem 3.)*

copyarray.c:

```
/*
 * CSE333 Midterm Solution Set, Spring 2012
 *
 * Copyright 2011 Steve Gribble
 */

#include <stdlib.h>
#include "copyarray.h"

int CopyArray(int *arr, int arrlen, int **outputarr) {
  // Be defensive; if arrlen is negative or zero, fail!
  if (arrlen <= 0)
    return 0;

  // Attempt to allocate the new array.
  int *newarr = (int *) malloc(arrlen*sizeof(int));
  if (newarr == NULL)
    return 0;

  // Do the copy.
  int i;
  for (i=0; i<arrlen; i++) {
    newarr[i] = arr[i];
  }

  // Set the output parameter, return success.
  *outputarr = newarr;
  return 1;
}
```

(***more space for problem 3.)***

copymain.c:

```
/*
 * CSE333 Midterm Solution Set, Spring 2012
 *
 * Copyright 2011 Steve Gribble
 */

#include <assert.h>
#include <stdlib.h>
#include <stdio.h>

#include "copyarray.h"

int main(int argc, char **argv) {
  // Test some corner cases.
  assert(CopyArray(NULL, -1, NULL) == 0);
  assert(CopyArray(NULL, 0, NULL) == 0);

  // Do a real test.
  int inarr[5] = {1, 2, 3, 4, 5};
  int *outarr, i;
  assert(CopyArray(inarr, 5, &outarr) == 1);
  for (i=0; i<5; i++) {
    assert(outarr[i] == inarr[i]);
  }
  free(outarr);

  return EXIT_SUCCESS;
}
```

Problem 4: Resplendent reference referendum (25 points)

Write the output of the following C++ code.

```
#include <stdlib.h>
#include <iostream>

int mystery(int *a, int &b, int c) {
  c++;
  b = b + c;
  *a = b;
  return *a - c;
}

int main(int argc, char **argv) {
  int w = 0, x = 1, y;
  y = mystery(&w, x, 5);

  int *a = &w;
  int &b = w;

  *a = mystery(&b, b, 6);

  std::cout << *a << " "<< b << " " << w << " ";
  std::cout << x << " " << y << std::endl;
  return 0;
}
```

7 7 7 7 1

Problem 5: A free point (1 point!)

Draw something fun…

(that's a band called "Fun." ☺)