CS 103: Introduction to Programming
Spring 2017 - Written Midterm Exam
2/23/17, 7PM – 8:30PM

Name: Answer Key + Rubric

Student ID: ______________________

Email: ____________________@usc.edu

Lecture section (Circle One):

Goodney 9:30am 11:00am 2:00pm

Page	Your score	Max score
2		11
3		9
4		6
5		6
6		12
7		10
8		8
Total		62

Note: The last page is blank and can be used for scratch paper.
Please turn it in with your exam

1. (9 pts.) Show what will be output by the cout's in this program. (Note: boolalpha simply causes Booleans to show on the screen as 'true' or 'false' when printed by cout):

```
#include <iostream>
#include <algorithm>
#include <cmath>

using namespace std;

int f1(int x, double y)
{
     return (int)(x*y);
}

int main()
{
     int x = 10, y=4;
     double z=10.5;
                                                                   1 pt ea
     cout << 15 % 7 << endl;                                 1
     cout << y/x << endl;                                    0
     cout << 5 + 3 % 2 * x / 4 << endl;                      7
     cout << (x++ + y) << endl;                              14
     cout << (++x - y) << endl;                              8
     cout << boolalpha << (!x && (y || y/x)) << endl;        false
     cout << pow(2, x) << endl;                              4096
     cout << max(min(max(x,y),x+y),(int)z) << endl;          12
     cout << f1(y,z) << endl;                                42
}
```

2. (2 pts.) a) Given the declaration: `char word[10];` What ***type*** will the expression `word` evaluate to? Circle your choice.

   a. char

   b. int*

   [✓ c.] char* +1

   d. char**

   b) What **value** will the expression `word` evaluate to?

   Starting address of the array
   address of word[0]
   or similar +1

3. (4 pts.) Given the following declarations and call to the function named **doit**, infer and write a correct **prototype** (i.e. pre-declaration of the function) for doit.

```
char c='C'; int X[10]; double z=3.14;
bool result = doit(c, X, &z);
```

Prototype for doit: +1 +1 +1

bool doit(char, int*, double*)

+1 or int[]

4. (5 pts.) Examine the program below (don't worry about #includes, etc.). What variables could legally be printed (i.e. would be in scope / accessible / visible) at each of the cout statements in the program. In the table below place a **check mark or x** in the corresponding cell if the variable **IS** in scope and can be printed by the corresponding cout. Leave the cell blank if the variable is NOT in scope for the given cout.

```
int a;
void f1(char c)
{  cout << ____ << endl; // cout1
}
int main()
{
  int i = 0;
  for(;i<10;i++){
      double z = pow(y, i);
  }
  cin >> a;
  if(a < 1){
      double y - 2.0/3.0;
  }
  cout << ____ << endl;  // cout2
}
```

	a	c	y	i	z
cout1	✓	✓			
cout2	✓			✓	

½ pt per box

5. (4 pts.) Given the following array declarations and indexed accesses, compute the address where the indexed value will be in memory. Assume the array starts at location 200 on a 64-bit computer.

   a. `int x[10]; x[5] is at:` 220 +1

   b. `char c[10][4]; c[2][1] is at:` 209 +1

   c. `double d[3][4][4]; d[1][2][3] is at:` 416 +1

   d. `char *n[10]; n[3] is at:` 224 +1

6. (2 pts.) Consider the following program fragment. The code is trying to find a random number n for which do_it() averages >= 3.5 for 10 iterations. do_it(n) always returns an integer in the range [0,5]. **Find and fix the problem by marking up the code.**

```
double average=0.0;
int n;
while(average < 3.5)
{
    int r=0; n = rand();
    for(int i=0;i<10;i++)
    {
        r += do_it(n);
    }
    average = (double)(r / 10);
}
```

+2

Anything to fix this cast:

(r*1.0)/10

(double)r/10

(double)(r/10.0)

etc.

7. (6 pts.) What does the following program output?

```
#include <iostream>

using namespace std;

int main()
{
      for(int i=0; i<10; (i<3?i++:i+=2) )
      {
            if( !(i%3))
            {
                  continue;
            }
            else if( i % 7 == 0)
            {
                  break;
            }
            cout << i << endl;

      }

}
```

Program Output:

1 +2

2 +2

5 +2

8. (12 pts.) Show the output of this program by tracing the code execution. For this code you can assume the first 10 calls to rand() returns the following sequence [18, 12, 8, 17, 1, 13, 10, 3, 16, 5]

```
#include <iostream>
#include <cstdlib>

using namespace std;

void m1(int *a, int *b)
{
      int c = *b;
      *b = *a;
      *a = c;
}

void m2(int *X, int n)
{
      for(int i=0;i<n-2;i++)
      {
            int j = rand() % (n-i);
            m1( (X+i) , X+i+j);
      }
}

int main()
{
      int array[] = {0,1,2,3,4};

      srand(1234);
      m2(array, 5);

      for(int i=0;i<5;i++)
            cout << array[i] << " ";
      cout << endl;

}
```

Output:

3 1 4 0 2

+2 +2 +2 +2 +2

+2 if 100% correct

9. (10 pts.) Show what will be printed by this program.

```
#include <iostream>
using namespace std;

void f1(int *d, unsigned char *s)
{
        int i-0;
        while(*s > 0)
        {
                (*(d + *s++ - 'a'))++;
        }
}

void f2(int *d, unsigned char *s)
{
        int i=0;
        while(*s != '\0')
        {
                while(*(d+i)>0)
                {
                        *s = 'a' + i;
                        d[i]--;
                        s++;
                }
                i++;
        }
}

int main()
{
        int data[] - {0,0,0,0};
        unsigned char str[]-"dacbdda";
        f1(data, str);
        for(int i-0;i<4;i++,cout << endl)
        {
                cout << (char)('a'+i) << ":" << data[i];
        }
        f2(data,str);
        cout << str << endl;
}
```

Program output: +1 +1 +1 +1

a:2 b:1 c:1 d:3

aabcddd +1

10. (8 pts.) Billy Bruin was attempting to write a program that would read in up to 16 ones or zeros of a binary number. The program calculates the value of the number and prints it out. For example if you enter 101x it will output "101=5" He has several questions and a few bugs. Can you help him out?

```
1   #include <iostream>
2   #include <cmath>
3
4   using namespace std;
5
6   int main()
7   {                                   +1
8           unsigned char str[17], c;
9           unsigned int val=0;
10          int i;
11          for(i=0;i<16;i++)
12          {
13                  cin >> c;   && +2
14                  if( c != '0' ~~||~~ c != '1')
15                  {
16                          break ; +1
17                  }
18                  str[i] = c;
19          }
20          str[i] = ∅  or '\0' or NULL  +2
21          int len = i;
22          for(i=0;i<len;i++)
23          {
24                  if( str[i] == '1')
25                  {
26                          val += pow(2, len - i);
27                  }
28          }
29          cout << str << "=" << val << endl;
30  }
```

a. Line 8: Billy wasn't sure how large to declare the array **str**. Please complete it for him.

b. Line 16: Billy wanted to immediately quit the for loop if he received something other than a '1' or '0'. Complete line 16.

c. After making these changes, Billy's first for-loop never seems to do anything. Can you fix it?

d. After that fix, Billy notices that the final cout sometimes prints extra garbage characters before the equals sign. Add something at line 20 to fix this.

e. Finally, Billy notices that his code seems to be **almost** working, except all of his answers are too big by a factor of two. Find the mistake and write the correction below:

Line num: 26 Changed code: val += pow(2, len-i-1); +2

Intentionally blank for scratch work. Please turn it in with your exam:

Name: ______________________________ Section time: __________