CS 103: Introduction to Programming
Fall 2016 - Written Midterm Exam
10/6/16, 7PM – 8:30PM

Name:________Solutions________________

Student ID: ____

Email: ______________________@usc.edu

Lecture section (Circle One):

Redekopp	(T/Th 9:30 a.m.)
Goodney	(M/W 2 p.m.) | (T/Th 11 a.m.) | (T/Th 12:30 p.m))

Page	**Your score**	**Max score**
2		12
3		9
4		5
5		5
6		12
7		8
8		9
Total		60

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

1. (10 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, int y)
{
  return (x+y)/2;
}

int main()
{
  int x = 16, y = 4, z=6;                                  // Answers
  cout << 34 % 6 << endl;                                  _____4______
  cout << 25 / 10 << endl;                                 _____2______
  cout << (5 + 9 % 4 * 2 / 3) << endl;                     _____5______
  cout << x++ << endl;                                     _____16______
  cout << --y << endl;                                     _____3______
  cout << boolalpha << (!(z < y) && (x != 0)) << endl;     _____true______
  cout << max(6, min(14, 5)) << endl;                      _____6______
  cout << pow(2, pow(2, f1(-2,6))) << endl;                _____16______
  return 0;
}
```

2. (2 pts.) Given the declaration: `int data[10];` What ***type*** will the expression `data` evaluate to? Circle your choice.
   a. int
   [✓ b. int*]
   c. int&
   d. int**

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.

```
int x; char word[10]; double z;
bool result = doit(&x, word, z);
```

Prototype for doit:

bool doit(int*, char*, double); or bool doit(int*, char[], double);

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=5;
void f1(char c)
{  cout << ____ << endl; // cout1
}
int main()
{
  double y;
  cin >> y;
  if(y < 2.0){
    int i = 0;
    while(i < 10){
      double z = pow(y, i);
      i++;
    }
  }
  f1('a');
  cout << ____ << endl;  // cout2
}
```

	a	c	y	i	z
cout1	x	x			
cout2	x		x		

5. (3 pts.) In the program below complete lines 4, 5, and 6 with any legal value such that the program will print out “USC > UCLA”

```
int x;
bool y;
bool z;

x = __1_______;  // anything other than 0
y = __true_______;
z = __true_______;

if(x) {

   if(!y) {
      cout << "UCLA > USC" << endl;
   }
   else if(!x) {
      cout << "UCLA > USC" << endl;
   }
   else if(z){
      cout << "USC > UCLA" << endl;
   }
   else{
      cout << "UCLA < USC" << endl;
   }
}
else
{
   cout << "UCLA > USC" << endl;
}
```

6. (2 pts.) Consider the following program fragment. Assume the function do_sim() returns an integer. The program is attempting to calculate the average value returned from do_sim() over many calls, however there is **one problem** that prevents a correct average from being computed. **Find and fix the problem by marking up the code**.

```
int num_sims=0;
cin >> num_sims;
int result=0;
for(int i=0; i<num_sims;i++)
{
   result += do_sim();
}
double average = (double) result / num_sims;  // cast either var. to a double
```

7. (5 pts.) What does the following program output?

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

int main()
{
      int x;
      for(x=1;x<8;x++)
      {
            if( (x % 4) == 0 )
            {
                  break;
            }
            else if( !(x % 2)  )
            {
                  continue;
            }
            cout << x << endl;
      }
      cout << x << endl;
}
```

Program Output:

```
1
3
4
```

8. (12 pts.) Show the output of this program by following the function call sequence. Carefully consider the arguments being passed:

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

void fa(int* x, double y)
{
  *x -= 1;
  y = 1.5;
  cout << "fa: " << *x << " " << y << endl;
}

void fb(int x, char* d, double* z)
{
  x = x-3;
  fa(&x, *z + 0.5);
  *d = 'b';
  cout << "fb: " << x << " " << *d  << " " << *z  << endl;

}

int main()
{
  int x = 103;
  char c = 'a';
  double y = 2.5;
  fb(x, &c, &y);
  cout << "main: " << x << " " << c << " " << y << endl;
  return 0;
}
```

Output:

```
f2: 99 1.5
f3: 99 b 2.5
main: 103 b 2.5
```

9. (8 pts.) Show what will be printed by this program assuming it is started with the following command line:

```
./prog 5 aaba dcaccdb
```

```
using namespace std;

void f1(int d[], char* s)
{
  cout << strlen(s) << " ";
  while( *s != '\0'){
    d[ *s - 'a' ]++;
    s++;
  }
  for(int i=0; i < 4; i++){
    while(d[i] > 0){
      --s;
      *s = 'a'+i;
      d[i]--;
    }
  }
  cout << s << endl;
}
int main(int argc, char* argv[])
{
  int x = atoi(argv[1]);
  cout << x << endl;
  for(int i=2; i < argc; i++){
    int dat[4] = {0,0,0,0};
    f1(dat, argv[i]);
  }
  return 0;
}
```

Program output:

```
5
4 baaa
7 ddcccba
```

10. (9 pts.) Billy Bruin was attempting to write a program that would read in up to 8 characters made of just ‘a’ or ‘b’ and then print out the letters with each one flipped to its opposite (i.e. a => b and b => a). Thus is the user typed in aabaabbb then the output should be bbabbaaa. If the user wanted to input less than 8 characters, they should type ‘q’ to quit. Thus the input: aabq should output just bba. Billy Bruin had some questions and made some mistakes. Answer Billy’s questions and identify the mistakes as described below the program:

```
1    #include <iostream>
2    using namespace std;
3
4    int main()
5    {
6      char str[_9___];
7      char temp;
8      int i;
9      for(i = 0; i < 8; i++){
10       cin >> temp;
11       if(temp == 'q'){
12         __break___;
13       }
14       if(temp == 'a'){
15         temp = 'b';
16       }
17       if(temp == 'b'){
18         temp = 'a';
19       }
20       str[i] = temp;
21     }
22     _str[i] = ‘\0’;  or str[i] = 0;______________________________
23     cout << str << endl;
24   }
```

a. Line 6: Billy wasn’t sure how large to declare the array **str**. Please complete it for him.

b. Line 12: Billy wanted to immediately quit the for loop if he received a ‘q’. Complete line 12.

c. Billy noticed good characters (a’s and b’s) being printed by the cout on line 23 but also some garbage characters after the a’s and b’s. Fix that problem by updating line 22.

d. Billy noticed that even with all the other problems fixed he was not seeing the correct a’s and b’s. What would Billy’s fixed program (from parts a-c) output for the input: **bbaq**. Show the output here: __aaa____________. Then describe what is wrong with his code and how you would fix it by modifying only 1 of the above lines. What line is the problem and how should it be change (write your answer below):

Line num: _17___ Changed code: else if(temp == ‘b’)