CS 103: Introduction to Programming
Fall 2015 - Written Midterm Exam
10/15/15, 7-8:30 PM

Name: Solutions
Student ID: 

Lecture section (Circle One): MW 12 | TTh 9:30 | TTh 11:00 | TTh 12:30 | TTh 5:00

<table>
<thead>
<tr>
<th>Page</th>
<th>Your score</th>
<th>Max score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>70</td>
</tr>
</tbody>
</table>
Question 1 [1 point]
What is the name of the function used to seed the random number generator?

srand

Question 2 [2 points]
Write a statement or two that returns a random number between 1 and 20 (inclusive). Assume all appropriate #include statements.

```c
int rand_between_1_and_20()
{
    // your code here
    return 1 + rand()%20;
    // -1 for rand()%19; -1 for num%rand(); -1 for forgetting 1+ (i.e. rand()%20)
}
```

Question 3 [2 points]
What does the following output?
```
for(int i=1;i<=6;i++) {
    cout << i/5 << endl;
} // for
```

0
0
0
0
1
1

Question 4 [2 points]
In the following, the array 's' is created on the stack. What is the statement that would cause 's' to be allocated on the heap/dynamically instead?

```c
void x() {
    double s[360];
    ...
    double *s = new double[360];
}
```

Question 5 [6 points]
Circle T or F (true or false) below. In C/C++:

T / False: typecasting a variable permanently changes the data type of that variable
T / False: every if must be followed by an else or an else if
**True** / F: it may be that the body of a while loop does not run even once
T / False: the ‘new’ operator allocates memory on the stack
**True** / F: the integer -1 is considered true
T / False: Using ‘>>’ to extract data from an input or file stream will read an entire line of input data

Question 6 [2 points]
What range of values can an 'unsigned char' type variable hold?

0 to 255;
Question 7 [2 points]
Given two programs a and b (runnable as ./a and ./b) that each read and write data via cin and cout, we would like to use the file x.dat to provide input to a, then pipe a’s output to b, and have b send its output to y.dat.

What would the command line for this be? Choose the correct answer below.

a. ./a > x.dat > ./b > y.dat
b. ./a | x.dat | ./b | y.dat
c. ./a < x.dat < ./b < y.dat
d. ./a < x.dat | ./b > y.dat
e. ./a > x.dat | ./b < y.dat
f. none of the above

Question 8 [5 points]
Write a function that returns the volume of a sphere with radius $r$ (given by the formula below). Assume the radius, $r$, is a ‘double’ and is the only argument provided to your function. Your answer (i.e. the volume) should be the return value. Don’t worry about #includes. Use the constant $\pi$ as the value of pi.

$$V = \frac{4}{3} \pi r^3$$

double volume(double r)
{
    return ( 4/3.0 * M_PI * pow(r,3));  // could be r*r*r rather than pow(r,3);
}

Question 9 [2 points]
Given the following, one way to change the 4th index in the ‘p’ array from 9 to -1 is shown. List two other different/alternate statements that would do the same.

```c
int p[5] = {1,3,5,7,9};
int *a = p;
```

Question 10 [2 points]
What is the output of the following program?

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

void fun(int x, int y) {
    x = 20;  y = 10;
}

int main() {
    int x = 10;
    fun(x, x);
    cout << x;
    return 0;
}
```

10
Question 11 [3 points]
What would the following output?

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

int main() {
    int a=10, b;
    b = a++ - 7 + a++;
    cout << a << " " << b << endl;
    return 0;
}
```

12 14

Question 12 [1 point]
Given the declaration:

```cpp
char name[] = "USC";
```

what does the expression `name` evaluate to? Be as specific as possible.

The starting address of the array; Pointer to the 0th character / 'U'
(Need some verbage about address/pointer to start/0th character of array)

Question 13 [4 points]
Given the declaration below:

```cpp
ifstream myfile("input.dat");
char data[100]; int x; double y;
```

What statements are syntactically correct (i.e. will compile)? Circle all that are correct.

a. `if (myfile.fail()) { cout << "Bad" << endl; }`
b. `myfile.getline(data, 100);`
c. `x = *myfile;`
d. `myfile >> x >> y;`
Question 14 [5 points]
Examine the following program - it is intended to fill an array of n integers with random numbers. List at least 4 compile or runtime errors in the program. List the line number and a brief explanation of what’s wrong and how it should be corrected.

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

int main()
{
    int n;
    srand(time(0));
    // Don’t worry about a prompt, the user will just enter a number
    cin << n << endl;

    int** data = new int[n];
    for(int i=0; i <= n; i++)
    {
        data[i] = srand();
    }
    delete [] data;
    return 0;
}
```

1.) Line 10: << arrows are the wrong way...should be >> for cin
2.) Line 10: Do not use `endl` w/ cin...remove to fix.
3.) Line 12: int** data should be int* data
4.) Line 13: i <= n should be i < n
5.) Line 14: srand() should be just rand()
6.) Missing `#include <ctime>`

Question 15 [6 point]
[12 pts.] In class and in labs you have learned about character arrays and used functions like ‘int strlen(char* arr)’, etc. In the space below complete the function ‘strCountLowerCase’ that should return a count of how many lower case letters exist in the character array, src. All other characters (uppercase, digits, spaces, etc.) should just be skipped. You may not call any other C/C++ library functions to do your work.

```cpp
int strCountLowerCase(char* src)
{
    int count = 0;
    int i=0;
    while( src[i] != '\0' ){
        if( src[i] >= 'a' && src[i] <= 'z') count++;
    }
    return count;
}
```
### Question 16 [8 pts.]
Examine the code below (line numbers are provided for easy referencing). Of the options listed below, circle the ones that are **POSSIBLE OUTPUTS** of the program for some value of \( x \) and \( y \) (i.e. do not circle outputs that are impossible for the program to produce).

**Possible outputs (circle all that apply):**

<table>
<thead>
<tr>
<th></th>
<th>ABD</th>
<th>ADE</th>
<th>ADEF</th>
<th>CD</th>
<th>CDEF</th>
<th>ACE</th>
<th>ACFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>int x, y;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>cin &gt;&gt; x &gt;&gt; y;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>if( x &gt; y ){</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>cout &lt;&lt; &quot;A&quot;;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>if( y &gt; 100){</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>cout &lt;&lt; &quot;B&quot;;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>else if( x &lt; 75 ){</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>cout &lt;&lt; &quot;C&quot;;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>if( y &lt; 75 ){</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>cout &lt;&lt; &quot;D&quot;;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>if( x == 20){</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>cout &lt;&lt; &quot;E&quot;;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>else if( y == 50){</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>cout &lt;&lt; &quot;F&quot;;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 point each

If B or C print, then A has to print.

Similarly if E or F prints then D has to print.

This rules out:

- CD, CDEF, ACE
- ABCD and ABD can't happen because y > 100 (B) and thus D can't happen.

ADE can't happen A means \( x > y \), D means \( y < 75 \) and E means \( x == 20 \), thus we know \( x \) is 20 and \( y < 20 \). But if that's the case we'd see C as well.

ADEF can't happen because E and F are mutually exclusive.

### Question 17 [8 pts.]

Analyze the following code and indicate what will be printed by the program in the box below.

```cpp
void f1(double r){
    r += 1.0;
    cout << r << endl;
}
void f2(double *s, int t){
    *(s+2) -= t;
    f1(s[0]);
    s++;
    cout << *s << endl;
    *s = 6.0;
}
int main() {  
    double s[4] = {1.5, 12.0, 23.5, 34.0};
    f2(s+1, 3);
    return 0;
}
```
Question 18 [9 points]
Show what this program outputs (in correct order) as it runs if invoked by the command line below. Notice there are two 'cout's in this program and each is inside a loop. Then add code to free all dynamic memory that this program allocated. Recall the first argument to strcpy() is the destination array followed by the source.

```
./prog abc
```

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

void f2(char** r, int d, char* w)
{
    int size, idx;
    for(int i=0; i < d; i++){
        size = d + 1 - i;
        r[i] = new char[size];
        strcpy(r[i], w+i);
        idx = 2*d-1-i;
        r[2*d-1-i] = new char[size];
        strcpy(r[idx], w+i);
        cout << i << " " << idx << " " << size << endl;
    }
}

int main(int argc, char* argv[])
{
    int len = strlen(argv[1]);
    char** r = new char*[2*len];
    f2(r, len, argv[1]);
    for(int i=0; i < 2*len; i++){
        cout << r[i] << endl;
    }

    // Write code to free up all dynamic memory
    for(int i=0; i < 2*len; i++)
        delete[] r[i];
    delete[] r;

    return 0;
}
```

0 5 4
1 4 3
2 3 2
abc
bc
c
c
bc
abc