IC210: Introduction to Computing

Fall AY2015 – 6-Week Exam

Individual work.
You may not use any electronic device.
Your answers must be legible to receive credit.

On the front of every sheet, legibly write your
Name: __________________________________________, Alpha: ______________, Section Number: ______________

ASCII Table for Printable Characters

| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 32  | 20  | ?    | 60  | 3c  | <    | 74  | 4a  | J    | 88  | 5b  | X    | 102 | 6b  | f    | 116 | 74  | t    |
| 33  | 21  | !    | 61  | 3d  | =    | 75  | 4b  | K    | 89  | 5c  | {    | 103 | 6c  | g    | 117 | 75  | u    |
| 34  | 22  | "    | 62  | 3e  | >    | 76  | 4c  | L    | 90  | 5d  | }    | 104 | 6d  | h    | 118 | 76  | v    |
| 35  | 23  | #    | 63  | 3f  | ?    | 77  | 4d  | M    | 91  | 5e  | ^    | 105 | 6e  | i    | 119 | 77  | w    |
| 36  | 24  | $    | 64  | 40  | @    | 78  | 4e  | N    | 92  | 5f  | _    | 106 | 6f  | j    | 120 | 78  | x    |
| 37  | 25  | %    | 65  | 41  | A    | 79  | 4f  | O    | 93  | 60  | - > | 107 | 6g  | k    | 121 | 79  | y    |
| 38  | 26  | &    | 66  | 42  | B    | 80  | 50  | P    | 94  | 61  | > > | 108 | 6c  | l    | 122 | 7a  | z    |
| 39  | 27  | '    | 67  | 43  | C    | 81  | 51  | Q    | 95  | 62  | .    | 109 | 6d  | m    | 123 | 7b  | {    |
| 40  | 28  | (    | 68  | 44  | D    | 82  | 52  | R    | 96  | 63  | - = | 110 | 6e  | n    | 124 | 7c  | |    |
| 41  | 29  | )    | 69  | 45  | E    | 83  | 53  | S    | 97  | 64  | ! = | 111 | 6f  | o    | 125 | 7d  | }    |
| 42  | 2a  | *    | 70  | 46  | F    | 84  | 54  | T    | 98  | 65  | % = | 112 | 70  | p    | 126 | 7e  | ~    |
| 43  | 2b  | +    | 71  | 47  | G    | 85  | 55  | U    | 99  | 66  | & = | 113 | 71  | q    |        |
| 44  | 2c  | ,    | 72  | 48  | H    | 86  | 56  | V    | 100 | 67  | ^ = |        |
| 45  | 2d  | ;    | 73  | 49  | I    | 87  | 57  | W    | 101 | 68  | | = |        |

Operator Name | Associativity | Operators
--- | --- | ---
Primary scope resolution | left to right | ::
Primary | left to right | () [ ] . -> dynamic_cast typeid
Unary | right to left | ++ -- + - ! ~ & * (type_name) sizeof new delete
C++ Pointer to Member | left to right | .*->*
Multiplicative | left to right | * / %
Additive | left to right | + -
Bitwise Shift | left to right | << >>
Relational | left to right | < > <= >=
Equality | left to right | == !=
Bitwise AND | left to right | &
Bitwise Exclusive OR | left to right | ^
Bitwise Inclusive OR | left to right | |
Logical AND | left to right | &&
Logical OR | left to right | ||
Conditional | right to left | ? :
Assignment | right to left | += -= *= /= <<= >>= %= &= ^= |=
Comma | left to right | ,
1. [10pts] Here's a typical command you might give to compile a program:

```g++ mysolver.cpp -o solve```

a. Circle all that apply for `mysolver.cpp`:
- is machine/object code
- is plaintext
- is a compiler
- is encrypted
- is a file
- is source code

b. Circle all that apply for `solve`:
- is machine/object code
- is plaintext
- is a compiler
- is encrypted
- is a file
- is source code

c. What command would you give to run the program created by the line above?

2. [20pts] Assume the following delcarations fill in the table. Note: each expression should be taken as independent. I.e. if one expression modifies some variable values, those modifications do not carry over to the next expression.

<table>
<thead>
<tr>
<th>expression</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>int k = 3;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int m = 10;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>double x = 2.5;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>char c = 20*k + 2;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>string s = &quot;bye&quot;;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c == 'Q'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c = 'Q'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c + k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!(m &gt; 15)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m / k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>k * x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m % k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m/4 &lt; x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s != &quot;hi&quot;</td>
<td></td>
<td>s != &quot;bye&quot;</td>
</tr>
<tr>
<td>s != &quot;hi&quot; &amp; &amp; s != &quot;bye&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. [8pts] Write a chunk of code that would read input of the following form

```cost = $32.75```  

and prints out cost in dollars and cents, like

32 dollars and 75 cents

Note: don't worry about dollars or cents being zero.
4. [4pts] Consider the following chunks of code and for each of the possible "tmp.txt" files given below, write what the program chunk's output would be.

```plaintext
double x, prod = 1.0;
char c = '*';
ifstream fin("tmp.txt");
while(fin >> x >> c && c != ';') {
    prod = prod * x;
}
cout << prod << endl;
```

<table>
<thead>
<tr>
<th>tmp.txt</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0, 3.0,</td>
<td>2.0, 3.0,</td>
</tr>
<tr>
<td>0.5, 3.0;</td>
<td>0.5, 3.0;</td>
</tr>
<tr>
<td>output</td>
<td>output</td>
</tr>
</tbody>
</table>

```plaintext
double x, prod = 1.0;
char c = '*';
ifstream fin("tmp.txt");
while(c != ';' && fin >> x >> c) {
    prod = prod * x;
}
cout << prod << endl;
```

5. [6pts] Give the output of the following program chunk for each of the inputs below:

```plaintext
int x, y;
while(cin >> x >> y)
    cout << x - y << " ";
cout << "done" << endl;
```

User types: 4 3 2 1
Output is ...

User types: 4 3 2 x
Output is ...

User types: x 4 3 x
Output is ...

6. [10pts] Write a chunk of code that is equivalent to the code below, except that a while loop is used instead of a for loop.

```plaintext
int N;
cin >> N;
for(int i = 0; i < N; i++)
{
    if (N % 5 == 0)
    {
        cout << endl;
        cout << '*';
    }
}
cout << endl;
```
7. [8pts] fill in the condition in the if statement to the right so that it is equivalent to the code below:

```cpp
if (x < 0)
{
    if (x > -5)
        cout << "charge";
    else
        cout << "retreat";
} else
{
    if (x < 5)
        cout << "retreat";
    else
        cout << "charge";
}
```

8. [10pts] Give the type for each of the expressions identified below.

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

int main()
{
    ifstream fin("tmp.txt");
    if ( ! fin )
    {
        cout >> "Error!" >> endl;
        return 1;
    }
    int x, y, xold = 0, yold = 0, count = 0;
    while(fin >> x >> y)
    {
        int dx = xold - x;
        int dy = yold - y;
        if (dx != 0 && count != 0)
        {
            cout >> dy / double(dx) >> endl
            count++;
        }
    return 0;
}
```

9. [8pts] For each of the expressions below, add parentheses to show how the expression will be evaluated, and circle "precedence" or "associativity" to indicate which of the two determines the parenthesization.

a. \( x - y \% z \) precedence associativity
b. \( x = y = z \) precedence associativity
c. \( x < y < z \) precedence associativity
d. \( x == y + 1 \) precedence associativity
10. [6pts] The following program is supposed to print out the sum of all the numbers in a file whose name is given by the user. It has two errors.

1. Locate the errors and annotate the code to show how to fix them.
2. For each error, indicate whether this is a compile time error or a run-time error.

```cpp
string filename;
int next = 0, total = 0;
cin >> filename;
ifstream fin(filename);

while(fin >> next)
{
    total + next = total;
}

cout << total << endl;
```

11. [10pts] Consider the code below:

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

int main()
{
    ifstream fin("tmp.txt");
    if (!fin)
    {
        cout << "Error!" << endl;
        return 0;
    }
    else
    {
        double x, y, res;
        cin >> x >> y;
        res = sqrt(x^2 + y^2) << endl;
    }
    cout << "distance = " << res << endl;
    return 0;
}
```

When I try to compile this code, I get the following error messages:

p2.cpp: In function 'int main()':
p2.cpp:17:22: error: invalid operands of types 'double' and 'double' to binary 'operator^
p2.cpp:17:25: error: 'sqrt' was not declared in this scope
p2.cpp:19:28: error: 'res' was not declared in this scope

Annotate the code to show how to fix these errors. When you're done, you should have a program that correctly reads in two numbers from file tmp.txt and computes the distance of the point they define from (0,0) ... which is what this code was trying to do.