IC210: Introduction to Computing

Fall AY2016 – 6-Week Exam

You may not use any electronic device.

This is a multi section exam that will be given to different midshipmen at different times. As per USNA INST 1531.53A, you may NOT communicate about this exam with anyone using any medium until your instructor tells you that you can.

Name: __________________________________, Alpha: __________, Section Number: __________
Instructor name: ____________________________________________

<table>
<thead>
<tr>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
<th>Dec Hex Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 20</td>
<td>46 2e</td>
<td>60 3c</td>
<td>74 4a</td>
<td>88 58</td>
<td>X</td>
<td>102 66</td>
<td>f</td>
</tr>
<tr>
<td>33 21 !</td>
<td>47 2f /</td>
<td>61 3d =</td>
<td>75 4b</td>
<td>K</td>
<td>89 59 Y</td>
<td>103 67 g</td>
<td>117 75 u</td>
</tr>
<tr>
<td>34 22 *</td>
<td>48 30 0</td>
<td>62 3e &gt;</td>
<td>76 4c L</td>
<td>90 5a Z</td>
<td>104 68 h</td>
<td>118 76 v</td>
<td></td>
</tr>
<tr>
<td>35 23 #</td>
<td>49 31 1</td>
<td>63 3f ?</td>
<td>77 4d M</td>
<td>91 5b [</td>
<td>105 69 l</td>
<td>119 77 w</td>
<td></td>
</tr>
<tr>
<td>36 24 $</td>
<td>50 32 2</td>
<td>64 40 @</td>
<td>78 4e N</td>
<td>92 5c \</td>
<td>106 6a j</td>
<td>120 78 x</td>
<td></td>
</tr>
<tr>
<td>37 25 %</td>
<td>51 33 3</td>
<td>65 41 A</td>
<td>79 4f 0</td>
<td>93 5d )</td>
<td>107 6b k</td>
<td>121 79 y</td>
<td></td>
</tr>
<tr>
<td>38 26 &amp;</td>
<td>52 34 4</td>
<td>66 42 B</td>
<td>80 50 P</td>
<td>94 5e ^</td>
<td>108 6c l</td>
<td>122 7a z</td>
<td></td>
</tr>
<tr>
<td>39 27 '</td>
<td>53 35 5</td>
<td>67 43 C</td>
<td>81 51 Q</td>
<td>95 5f _</td>
<td>109 6d m</td>
<td>123 7b {</td>
<td></td>
</tr>
<tr>
<td>40 28 (</td>
<td>54 36 6</td>
<td>68 44 D</td>
<td>82 52 R</td>
<td>96 60 `</td>
<td>110 6e n</td>
<td>124 7c</td>
<td></td>
</tr>
<tr>
<td>41 29 )</td>
<td>55 37 7</td>
<td>69 45 E</td>
<td>83 53 S</td>
<td>97 61 a</td>
<td>111 6f o</td>
<td>125 7d }</td>
<td></td>
</tr>
<tr>
<td>42 2a *</td>
<td>56 38 8</td>
<td>70 46 F</td>
<td>84 54 T</td>
<td>98 62 b</td>
<td>112 70 p</td>
<td>126 7e ~</td>
<td></td>
</tr>
<tr>
<td>43 2b +</td>
<td>57 39 9</td>
<td>71 47 G</td>
<td>85 55 U</td>
<td>99 63 c</td>
<td>113 71 q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 2c :</td>
<td>58 3a :</td>
<td>72 48 H</td>
<td>86 56 V</td>
<td>100 64 d</td>
<td>114 72 r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45 2d -</td>
<td>59 3b ;</td>
<td>73 49 I</td>
<td>87 57 W</td>
<td>101 65 e</td>
<td>115 73 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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. [9pts] Suppose you (successfully) give the command: `g++ prog1.cpp -o george`

   a. Which of these is an executable program (circle all that apply): `g++ prog1.cpp george`
   b. Which of these is a file (circle all that apply): `g++ prog1.cpp george`
   c. Which of these is created by the command (circle all that apply): `g++ prog1.cpp george`

2. [20pts] Assuming the following definitions, 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.

   ```
   bool b = true;
   int n = 0;
   int j = 6;
   double z = 1.5;
   char c = 67;
   ifstream fin("foo"); // assume foo exists in 
   // the current directory
   ```

<table>
<thead>
<tr>
<th>expression</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>c == 'c'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c = 10*j + z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int('3' * 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b = n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/j + z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>j * z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!fin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>j % 3 &amp; &amp; b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 + j/10 &gt; z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n != 0</td>
<td></td>
<td>n != 10</td>
</tr>
</tbody>
</table>

3. [8pts] Programs are often given version numbers in the form `vmajorVersion.minorVersion`. A higher majorVersion number means a more recent version of the program. When the majorVersion numbers are the same, the larger minorVersion number denotes the more recent program. So, v3.2 is more recent than v1.8, and v3.12 is more recent than v3.9 (because 12 is bigger than 9).

   Fill in the below to create a program that reads in a version number (e.g. `v5.24`) and writes the word "newer" if the version is more recent than v2.7 and "not newer" otherwise. Here are some example runs:

   ```
   $ ./go
   v1.8
   not newer
   $ ./go
   v2.7
   not newer
   $ ./go
   v2.15
   newer
   $ ./go
   v3.0
   newer
   ```

   ```
   #include <iostream>
   using namespace std;
   int main()
   {

   return 0;
   }```
4. [8pts] The code below reads in values $i$, $j$ and $k$, then prints out the value of $i \times j^k$. Write a chunk of code that is equivalent to the code below, except that a while loop is used instead of a for loop.

```cpp
int i, j, k, e = 1;
    cin >> i >> j >> k;

for(int i = 0; i < k; i++)
    e = e*j;

    cout << i << '*' << j << '^' << k
    << " = " << i*e << endl;
```

5. [8pts] Consider sequences like this

d23 d2 u101 d5 u6 d19 x

representing a sequence of down (d) and up (u) moves. The x just terminates the sequence. Fill in the blank in the following code to make a program that reads a sequence like the above from the user (assume valid input, i.e. don't worry about error checking!) and prints out the sum of all the "down" moves, followed by the sum of all the "up" moves. A sample run would look like this

```
u9 d2 d7 u3 d1 x    ← user input
down = 10
up = 12
```

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

int main()
{
    int up = 0, down = 0;

    cout << "down = " << down << endl;
    cout << "up = " << up << endl;
    return 0;
}
```
6. [9pts] Fill in the conditions on the if statements on the right so that they are equivalent to the code on the right.

a. 
```cpp
if (y == 0 || x == 0) {
    // do nothing, avoids divide by zero
} else {
    cout << 1/x + 1/y;
}
```

b. 
```cpp
if (w % 2 == 0) {
    if (d == 6) {
        b += p;
    }
}
```

c. 
```cpp
if (a <= 'k') {
    cout << "easy";
} else {
    if (a >= 'q') {
        cout << "easy";
    } else {
        cout << "tough";
    }
}
```

7. [8pts] For the following questions you might want to consult the operator precedence/associativity table on the front of this exam.

a. Is \( x = y \neq z \) evaluated as \( (x = y) \neq z \) or as \( x = (y \neq z) \) ? (circle one)

   Explain your answer!

b. Is \( x == y \neq z \) evaluated as \( (x == y) \neq z \) or as \( x == (y \neq z) \) ? (circle one)

   Explain your answer!
8. [10pts] The following program is supposed to read in a starting number from the user (we can assume a positive number) and countdown by two's from the starting number. It sometimes works, for example if you input 10 it outputs

10 8 6 4 2 blast off!

which is what I want. But it sometimes seems to get stuck in an infinite loop. For example, if you input 11 you get an infinite loop rather than what I was hoping for:

11 9 7 5 3 1 blast off!

a. Annotate the code to show how to fix this problem.

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

int main()
{
    cout << "Enter start: ";
    int k;
    cin >> k;
    for(int i = k; k != 0; k = k - 2)
    {
        cout << k << " ";
    }
    cout << "blast off!" << endl;
    return 0;
}
```

b. Explain why the original fails sometimes, but not always.

9. [10pts] Give the type for each of the expressions identified below.

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

int main()
{
    double bal = 700, pay = 550;
    double rate = 3.1, expense = 45;
    int week;
    cin >> week;
    for(int day = 1; day <= week * 7; day++)
    {
        if (day % 14 == 13)
        {
            bal += pay;
        }
        bal -= expense;
        bal += bal*rate/1200;
    }
    string m = "You are ";
    if (bal < 0)
    {
        m += "bankrupt";
    } else
    {
        m += "solvent";
    }
    cout << m << endl;
    return 0;
}
```
10. [10pts] When I try to compile the code below, I get the following error messages:

d9.cpp: In function 'int main()':
d9.cpp:11:18: error: no match for 'operator<<' in 'fin << x'
d9.cpp:13:19: error: lvalue required as left operand of assignment
d9.cpp:19:5: error: 'total' was not declared in this scope
d9.cpp:21:11: error: 'total' 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 either prints out -1, if file tmp.txt does not exist, or prints the sum of all the even numbers in tmp.txt, assuming it's just a file full of integers.

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

int main()
{
    ifstream fin("tmp.txt");
    if (fin)
    {
        int total = 0, x;
        while(fin << x)
        {
            if (x % 2 == 0)
                total = total + x;
        }
    } else
    {
        total = -1;
    }
    cout << total << endl;
    return 0;
}
```