

Name: _____ Alpha _____ Instructor: _____

This is a SAMPLE exam – not a comprehensive summary of topics! Use this to gauge the types of questions that may be asked, but review all your notes and problems!

**IC210
Sample Written 12-Week Exam**

This is a multi section exam. You may NOT communicate about this exam with anyone using any medium until your instructor tells you that you can.

This exam is closed book and no electronic device can be used.

Problem	Points Avail	Score
1.	15	
2.	15	
3.	15	
4.	20	
5.	15	
6.	20	
Total	100	

1. (15 pts) Indicate **exactly** what prints at each comment.

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

int doSomething(int, int&);
int main()
{
    int a, b, c;
    a = 4;
    b = 7;

    cout << "a=" << a << " b=" << b << endl;
    // What is printed by the above line?
```

```
c = doSomething(a*3, b);
cout << "a=" << a << " b=" << b << " c=" << c << endl;
// What is printed by the above line?
```

```
return 0;
}
```

```
int doSomething(int x, int &y)
{
    x = x + y;
    y = 3 * y;
    cout << " x=" << x << "y=" << y << endl;
    // What is printed by the above line?

    return x;
}
```

2. (15 points) In answering the following questions, if the function definitions are not explicitly given, assume that the correct definitions for the given prototypes do appear somewhere later in the code.

a. Given the following prototype and code fragment for `main()`, why does the compiler complain about the function call to `simpleFunc` in `main()`?

```
void simpleFunc(int &num, int &denom);

int main()
{
    double x;
    cin >> x;
    int y = x * 12;
    simpleFunc(12, y);

    return 0;
}
```

b. Why does this print zero no matter what the user enters?

```
void readpos(int n);

int main()
{
    int k = 0;
    readpos(k);
    cout << k << endl;
    return 0;
}

void readpos(int n)
{
    do
    {
        cout << "Enter a positive integer: ";
        cin >>n;
    } while (n <=0);
}
```

c. When I run the following program it halts with an error...why?

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

int fact(int n);

int main()
{
    int n;
    cout << "Enter a number: ";
    cin >> n;
    cout << n << " != " << fact(n) << endl;

    return 0;
}

int fact (int n)
{
    if (n == 0)
        return 1;
    else
        return n*fact(n);
}
```

3. (15 pts) Provide the prototypes (not the definitions) for the following functions.

Description & Example Code	Prototype
<p>ending, which takes an array of strings and returns an array of chars containing the last character of each string. Example of use:</p> <pre> int n= 10; string *s = new string[n]; for (int i = 0; i < n; i++) cin >> s[i]; char *f = ending(s, n); cout << f[2] << " is the last " << "letter of " << s[2] << endl; </pre>	
<p>divBy2_3, which examines an array and determines the number of elements divisible by 2 and the number of elements divisible by 3. Examples of use:</p> <pre> int n = 50, div2, div3; int *a = new int[n] for (int i=0; i<n; i++) cin >> a[i]; divBy2_3(a, n, div2, div3); cout << "There are " << div2 << " divisible by 2 and " << << div3 << " divisible by 3 " << "integers in the array." << endl; </pre>	
<p>read, which reads an array of ints from either a file or the keyboard. Example of use:</p> <pre> int n = 10; int *a = new int[n]; string s; if (cin >> s && s == "keyboard") read (a, n, cin); else { ifstream fin (s.c_str()); read (a, n, fin); } </pre>	

4. (20 points) A function `lessThan` accepts four arguments: a pointer to an array of integers, an integer indicating the size of the array, a double value and an integer. The number of elements in the array that are less than or equal to the double value that is passed from the function back to the calling program via the last integer parameter. The function also explicitly returns the number of elements in the array that are equal to zero. Given the following declarations

```
int n = 10;
int *a = new int [n];
double value = 12.5;
int larger, smaller;
```

(a) (5 pts) Give the prototype for the function `lessThan`.

(b) (5 pts) Give the function call statement as it would appear in `main()` including all necessary variable declarations.

(c) (10 pts) Write the function `lessThan`. You may assume that `main` has filled the array with values before it calls `lessThan`.

5. (15 pts) The following function named `log` takes two integer arguments (`a` and `b`) such that `a` is an exact power of `b`, returns its exponent value. For example, if `a = 81` and `b = 3`, then the function call `log (a, b)` returns 4 since $3^4 = 81$. Assume `a` is always an exact power of `b`. An iterative version of the function is shown below. In the space provided, write a recursive function that accomplishes the same task.

```
int log(int a, int b)
{
    int power = 0;
    while (a >= b)
    {
        power++;
        a = a/b;
    }
    return power;
}
```

The equivalent recursive function is:

