IC210: Introduction to Computing

Fall AY2015 – 12-Week Exam

Individual work.
Closed book. Closed notes.
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
32	20		46	2e	.	60	3c	<	74	4a	J	88	58	X	102	66	f	116	74	t
33	21	!	47	2f	/	61	3d	=	75	4b	K	89	59	Y	103	67	g	117	75	u
34	22	"	48	30	0	62	3e	>	76	4c	L	90	5a	Z	104	68	h	118	76	v
35	23	#	49	31	1	63	3f	?	77	4d	M	91	5b	[	105	69	i	119	77	w
36	24	$	50	32	2	64	40	@	78	4e	N	92	5c	\	106	6a	j	120	78	x
37	25	%	51	33	3	65	41	A	79	4f	O	93	5d	]	107	6b	k	121	79	y
38	26	&	52	34	4	66	42	B	80	50	P	94	5e	^	108	6c	l	122	7a	z
39	27	'	53	35	5	67	43	C	81	51	Q	95	5f	_	109	6d	m	123	7b	{
40	28	(	54	36	6	68	44	D	82	52	R	96	60	`	110	6e	n	124	7c	|
41	29	)	55	37	7	69	45	E	83	53	S	97	61	a	111	6f	o	125	7d	}
42	2a	*	56	38	8	70	46	F	84	54	T	98	62	b	112	70	p	126	7e	~
43	2b	+	57	39	9	71	47	G	85	55	U	99	63	c	113	71	q			
44	2c	,	58	3a	:	72	48	H	86	56	V	100	64	d	114	72	r			
45	2d	-	59	3b	;	73	49	I	87	57	W	101	65	e	115	73	s			

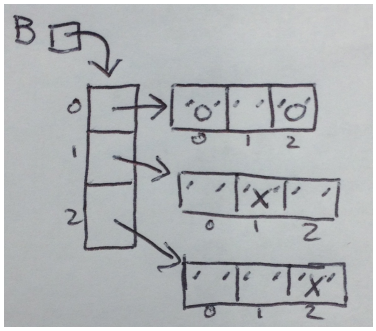
Operator Name	**Associativity**	**Operators**
Primary scope resolution	left to right	::
Primary	left to right	() [ ] . -> dynamic_cast typeid
Unary	right to left	++ -- + - ! ~ & * (*type_name*) sizeof new delete
C++ Pointer to Member	left to right	.*->*
Multiplicative	left to right	* / %
Additive	left 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. [15pts] In the following code, clearly identify (e.g. circle and label) every

   1. function prototype
   2. function definition
   3. function call
   4. function argument
   5. function parameter

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

int readLength();

void writeLength(int x);


int main()
{
  cout << "Length of ladder: ";

  double L = readLength();


  cout << "Height of gutters: ";

  double H = readLength();


  double w = asin(H/L);

  double D = L*cos(w);


  cout << "Distance from house: ";

  writeLength(D);

  cout << endl;

  return 0;
}


int readLength()
{
  int f, i;

  char c;

  cin >> f >> c >> i >> c;

  return 12*f + i;
}


void writeLength(int x)
{
  int f = x/12, i = x%12;

  cout << f << '\'' << ' ' << i << '"';
}
```

2. [10pts] Write the code (as it would appear in main(), for example) that creates a variable B and allocates and initializes so that we have the situation depicted in the following picture:



3. [9pts] Consider the following main() function:

```
int main()
{
  double* A = mkFilledToN(15);  // Creates array containing 1,2,...,n
  addXThenSine(A,15,0.3);    // Replaces each A[i] with sin(A[i] + .3)
  double minA, maxA;
  getMinMax(A,15,minA,maxA); // finds min and max elements of A
  cout << minA << ' ' << maxA << endl;
  return 0;
}
```

a. Write the prototype for mkFilledToN:

b. Write the prototype for addXThenSine:

c. Write the prototype for getMinMax:

4. [6pts]

function foo	function main

int foo(int x, int &y)
{
  x = x * 2;
  y = y * 3;
  int results = y * x;
  return results;
}
	int main()
	{
	  int x = 4;
	  int y = 5;
	  int z = foo(x,y);
	  ← value of x is _____
	  value of y is _____
	  value of z is _____

5. [7pts] Consider the following code fragment:

```
int * A = new int[10];
int n;
cin >> n;
if (n > 10)
{
  A = new int[n];
}
A[0] = 42;
```

Assume that the user enters **15**, so that the cin reads 15.

a. by the last line, A points to an array of size: 10, 15, neither, A out of scope (circle one)

b. what happens to the array(s) that A no longer points to?

6. [10pts] Give the prototype and definition for a function swap such that the code below prints the reverse of what the user entered. E.g. user enters 7 5, program prints 5 7.

```
int a, b;
cin >> a >> b;
swap(a,b);
cout << a << ' ' << b << endl;
```

7. [20pts] Assuming the following function prototypes and variable definitions, fill in the table. If the expression is a syntax error, write "error" for BOTH type and value. **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.

```
// Function prototypes - each of these functions does what the name says it does
int max(int a, int b); // returns the larger of a and b
bool ss(double x, double y); // returns true if x and y both positive, both negative or both zero
                             // returns false otherwise
int blank(string &s); // returns length of s, and sets all characters in s to ' ', i.e. to space.

// Variable definitions
int k = 2, m = 7;
double x = 2.8;
string s = "four";
int *A = new int[3];  A[0] = 3; A[1] = -3; A[2] = 0;
double *V = new double[2]; V[0] = 2.7; V[1] = -2.3;
```

expression	type	value
max(k,m)/2		
max(k,s)/2		
blank(s) > 0 && s[0] == ' '		
ss(V[0],V[1])		
max(blank(s),2)		
max(A[0]-2,V[0])		
max(5,3) + V[0]		
ss(0.2,0.5) && V[0] > k		
blank(s + "boo")		
ss(-0.2,-0.3) && max(1,0)		

8. [12pts] Consider the following function definition, along with prototypes for two functions it calls. Identify the types of the indicated expressions. Note: there's no reason to try to understand what this function does. That's not what this question is about. It's just about identifying types of expressions in real code.

```
bool findit(string* A, int n);
char* mkit(int k);

int subit(string** M, int r, int c)
{
  char* R = mkit(r-1);

  for(int i = 0; i < r; i++)
  {
    if (i != r - 1 && findit(M[i],c))
      R[i] = M[i+1][0][0];
  }

  int t = 0;

  for(int j = 0; j < r-1; j++)

    if (65 <= R[i] && R[i] <= 90)
      t++;

  return t;
}
```

a. _____

b. _____

c. _____

d. _____

e. _____

f. _____

9. [8pts] Consider the code below:

```
 1 #include <iostream>
 2 #include <fstream>
 3 #include <cmath>
 4 using namespace std;
 5
 6 string* getWords(istream in, int k);
 7
 8 int main()
 9 {
10   ifstream fin("tmp.txt");
11   string *A = getWords(fin,10);
12
13   for(int i = 0; i < 10; i++)
14   {
15     if ("A" <= A[i][0] && A[i][0] <= "Z")
16       A[i][0] += 32;
17   }
18
19   for(int i = 0; i < 10; i++)
20     cout << A[i] << endl;
21
22   return 0;
23 }
24
25 string* getWords(istream in, int k)
26 {
27   string *W = new string*[k];
28   while(--k >= 0 && in >> W[k]);
29   return W;
30 }
```

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

```
ex4.cpp:6:9: error:   initializing argument 1 of 'std::string* getWords(std::istream, int)'
ex4.cpp:15:22: error: ISO C++ forbids comparison between pointer and integer [-fpermissive]
ex4.cpp:15:38: error: ISO C++ forbids comparison between pointer and integer [-fpermissive]
ex4.cpp:27:28: error: cannot convert 'std::string**' to 'std::string*' in initialization
```

Annotate the code to show how to fix these errors. Note: the program is supposed to read in the first 10 words of the file tmp.txt, change any leading capital letters to lower case, and then print the words out. Your fixes shouldn't change that!

10. [3pts] The image below is a deadmau5 CD cover. What's the joke behind the album title "while(1<2)"?