1. Print your name on your scantron in the space labeled NAME.
2. Print CMSC 4023 in the space labeled SUBJECT.
3. Print the test number and version, T2/V1, in the space labeled TEST NO.
4. Print the date, 10-30-2013, in the space labeled DATE.
5. Print your CRN number, 12105, in the space labeled PERIOD.
6. This is a closed-book examination. No reference materials are permitted. No notes are permitted.
7. You may not consult your neighbors, colleagues, or fellow students to answer the questions on this test.
8. Cellular phones are prohibited. The possessor of a cellular phone will receive a zero (0) if the phone rings or is visible during the test.
9. Mark the best selection that satisfies the question. If selection b is better that selections a and d, then mark selection b. Mark only one selection.
10. Darken your selections completely. Make a heavy black mark that completely fills your selection.
11. Answer all 25 questions.
12. Record your answers on SCANTRON form 882-E (It is green!)
1. (Ch. 5. p 222) Select the correct storage type for variable $m$.

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

int max(int A[], int N)
{
    int m = A[0];
    for (int i = 1; i < N; i++) if (A[i] > m) m = A[i];
    return m;
}

int main()
{
    int A[] = {17, 2, -3, 11, 5, -7, -13};
    cout << max(A, 7) << endl;
    return 0;
}
```

Figure 1 Program for Question 1

a. Static variable
b. Stack-Dynamic variable
c. Explicit Heap-Dynamic variable
d. Implicit Heap-Dynamic variable

2. (Ch. 5. p. 233) What is printed by the program in Figure 2 assuming dynamic scope?

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

int a, b;

void print(void) { cout << "a=" << a << " b=" << b; }

int p(int& a) { int p = 2; a = 0; b = 1; return p; }

void q(void) { int b = 4; a = 3; print(); }

int main() { a = p(a); q(); return 0; }
```

Figure 2 C++ Program for Question 2

a. a=1 b=3
b. a=2 b=4
c. a=3 b=1
d. a=3 b=4
3. (Ch. 5 p. 225) What is printed by the program in Figure 3 assuming static scope?

```cpp
#include <iostream>
using namespace std;
int a,b;
void print(void){cout << "a=" << a << " b=" << b;}
int p(int& a){int p=2;a=0;b=1;return p;}
void q(void){int b=4;a=3;print();}
int main(){a=p(a);q();return 0;}
```

Figure 3 C++ Program for Question 3

a. a=3 b=4
b. a=2 b=4
c. a=1 b=3
d. a=3 b=1

4. (Ch. 5. p 216) Identify the binding type illustrated by the JavaScript statements shown in Figure 5.

```javascript
list = [10.2,3.5];
list=47;
```

Figure 5 C++ Program for Question 5

a. dynamic
b. static
c. inferential
d. explicit
5. (Ch. 5. p 222) What is the lifetime of variable \( p \) declared on line 6 of Figure 5?

```cpp
#include <iostream>
using namespace std;
int a;
int b;
int p(int& a)
{
 int p=2;
 a=0;
 b=1;
 return p;
}

void q(void)
{
 static int b=4;
 a=3;
}

int main()
{
 a=p(a); q();
 return 0;
}
```

**Figure 5 Program for Question 5**

a. Storage for variable \( p \) is allocated when function \( \text{main} \) is called and its storage is reclaimed when function \( \text{main} \) returns.

b. Storage for variable \( p \) is allocated during translation and reclaimed when the file containing executable form of the program in figure 5 is deleted.

c. Storage for variable \( p \) is allocated when function \( p \) is called and its storage is reclaimed when function \( p \) returns.

d. Storage for variable \( p \) is allocated at load time and reclaimed when the program in Figure 5 returns control to the operating system.

6. (Ch. 5. p. 214) When is the range of possible values for a variable of type \( \text{int} \) assigned in the programming language C++?

a. Execution time

b. Translation time

c. Language definition time

d. Language implementation time

7. (Ch. 6. p. 259) Which of the following declarations does NOT define an ordinal type?

a. `enum color {red, green, blue};`

b. `type day is ordinal (Mon, Tue, Wed, Thu, Fri, Sat, Sun);`

c. `subtype Index is Integer range 1..100;`

d. `char`
8. (Ch. 6. p. 277) Assume that each element of array $A$ occupies $e$ bytes, array $A$ has $r$ rows whose indexes are 1, 2, ..., $r$ and $c$ columns whose indexes are 1, 2, ..., $c$. Array $A$ is allocated in row-major order. Without subscripts $A$ is the address of the first byte allocated. Which of the following expressions gives the address of the first byte of element $A[i][j]$?

a. $A + ((i - 1) \cdot c + j - 1) \cdot e$

b. $A + i \cdot c \cdot e + j \cdot e$

c. $A + ((j - 1) \cdot c + i - 1) \cdot e$

d. $A + j \cdot c \cdot e + i \cdot e$

9. (Ch. 6. p. 256) Which of the following is NOT a string length option discussed in our text?

a. static length string

b. limited dynamic length string

c. dynamic length string

d. null-terminated string

10. (Ch. 6. p. 307) Which kind of type checking makes type $X$ equivalent to type $Y$?

```plaintext
type X=array[1..10] of integer; type Y=array[1..10] of integer;
```

a. dynamic

b. referential

c. structural

d. static
11. (Ch. 6. p. 288) Which set of C++ declarations most closely represents the Ada declarations given in the diagram below?

```cpp
#include <iostream>

using namespace std;

enum Shape {Circle, Triangle, Rectangle};
enum Colors {Red, Green, Blue};

struct Figure {
    float Diameter;
    int Left_Side;
    int Right_Side;
    float Angle;
}

struct Triangle {
    int Left_Side;
    int Right_Side;
    float Angle;
}

struct Rectangle {
    int Side_1;
    int Side_2;
}

struct Circle {
    float Diameter;
}

Figure circle = {Diameter: 5.0};
Figure triangle = {Left_Side: 3, Right_Side: 4, Angle: 90};
Figure rectangle = {Side_1: 5, Side_2: 6};
```

Figure 11  Declarations for Question 11
enum Shape {Circle, Triangle, Rectangle};
enum Colors {Red, Green, Blue};
struct Type_Triangle {
    int Left_Side, Right_Side;
    float Angle;
};
struct Type_Rectangle {
    int Side_1, Side_2;
};
struct Type_Circle {
    float Diameter;
};
struct Type_Form {
    Type_Circle C;
    Type_Triangle T;
    Type_Rectangle R;
};
struct Figure {
    bool Filled;
    Colors Color;
    Shape Form;
    Type_Form F;
};
```c
enum Shape {Circle, Triangle, Rectangle};
enum Colors {Red, Green, Blue};
union Type_Form {
    int Side_1, Side_2;
    float Diameter;
    int Left_Side, Right_Side; float Angle;
};
struct Figure {
    bool Filled;
    Colors Color;
    Shape Form;
    Type_Form F;
};
```

```c
enum Shape {Circle, Triangle, Rectangle};
enum Colors {Red, Green, Blue};
struct Type_Triangle {
    int Left_Side, Right_Side;
    float Angle;
};
struct Type_Rectangle {
    int Side_1, Side_2;
};
struct Type_Circle {
    float Diameter;
};
struct Type_Form {
    Type_Circle C;
    Type_Triangle T;
    Type_Rectangle R;
};
union Figure {
    bool Filled;
    Colors Color;
    Shape Form;
    Type_Form F;
};
```
12. (Ch. 6. p. 293) Which code fragment produces a dangling pointer?

```
#include <iostream>
using namespace std;
int* f(void)
{
    int d;
    return &d;
}
int main()
{
    int* p=f();
    return 0;
}
```

```
#include <iostream>
using namespace std;
int* f(void){return new int;}
int main()
{
    int* p=f();
    int q;
    p=&q;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int* p=new int;
    int* q=p;
    delete q;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int* f(void){return new int;}
int main()
{
    int* p=f();
    p=new int;
    return 0;
}
```

13. (Ch. 6. p. 306) Which feature of C++ is primarily responsible for characterizing the language as NOT strongly typed?

   a. mixed-mode coercion
   b. user-defined operator overloading
   c. polymorphic pointers
   d. union types

14. (Ch. 7. p. 324) According to Sebesta, what operator usually associates to the right?

   a. exponentiation operator **
   b. unary minus -
   c. assignment =
   d. prefix increment ++
15. (Ch. 7. p. 332) Which of the following code fragments contains an example of coercion?

```cpp
#include <string>
using namespace std;
int main()
{
    string s="tomat";
    char c='o';
    s=s+c;
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    double x=1.0;
    double y=2.0;
    int i=(int)(x+y);
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    char A=0x20;
    A = A << 1 | 0x01;
    cout << A << endl;
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    double x=1.0;
    double y=x+1;
    return 0;
}
```

**Figure 15 a**

**Figure 15 b**

**Figure 15 c**

**Figure 15 d**

16. (Ch. 7. p. 337) Identify the order of evaluation for the expression given in the Figure below?

Please note that expressions are evaluated in the order in which they appear from left to right. Expressions are separated by commas in the selections below.

```cpp
int a=0; int b=5;
a>=0 || b<5?a=5:b=0;
```

**Figure 16 Expression for Question 16**

a. $a>=0, b<5, a=5, b=0$

b. $a>=0, a=5$

c. $a>=0, b<5, a=5$

d. $a>=0, b<5, b=0$
17. (Ch. 7, p. 321) Assume the following rules of associativity and precedence for expressions.

<table>
<thead>
<tr>
<th>Operators</th>
<th>Precedence</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>*, /, not</td>
<td>highest</td>
<td>left to right</td>
</tr>
<tr>
<td>+, - &amp; mod</td>
<td></td>
<td>left to right</td>
</tr>
<tr>
<td>- (unary)</td>
<td></td>
<td>right to left</td>
</tr>
<tr>
<td>=, /=, &lt;, &lt;=, &gt;=, &gt;</td>
<td></td>
<td>left to right</td>
</tr>
<tr>
<td>and</td>
<td></td>
<td>left to right</td>
</tr>
<tr>
<td>or, xor</td>
<td>lowest</td>
<td>left to right</td>
</tr>
</tbody>
</table>

Show the order of evaluation of the expression in the Figure below by parenthesizing all sub-expressions and placing a superscript on the right parenthesis to indicate the order. For example, for the expression

\[ a + b \times c + d \]

the order of evaluation would be represented as

\[ (((a + b) \times c)^2 + d)^3 \]

\[
\frac{(a - b) \times e}{a - 3}
\]

**Figure 17 Expression for Question 17**

\[ (a - b) \times (d \times e) / (a - 3)^4 \]

a. \[ (((a - b)^3 / c)^2 \& (d \times e)^3 / (a - 3)^4)^5)^6 \]

b. \[ ((a - b)^3 / c \& (d \times e)^3 / (a - 3)^2)^4)^5)^6 \]

c. \[ (((a - b)^3 / c \& ((d \times e)^3 / a)^2 - 3)^4)^5 \]

d. \[ (((d \times e) / a - 3)^2 / c)^5 \& (((d \times e)^2 / a)^3 - 3)^4)^6 \]
18. (Ch. 7, p. 321) Mark the selections that satisfy the relation given in Figure 18.1 for the grammar of Figure 18.2. Function \( p(op) \) returns the precedence of the argument \( op \). Argument \( op \) is an operator in the grammar given in Figure 18.2. The operators given in the grammar of Figure 18.2 include +, -, unary -, *, /, and ^. The operator, unary-, appears in production 8.

\[
p(op_1) \geq p(op_2) \geq \cdots \geq p(op_n) \mid op_i \in \{ +, -, /, ^, \text{unary-}\}
\]

Figure 18.1

```
1    expression  \rightarrow  term
2    expression  \rightarrow  term addop expression
3    term        \rightarrow  factor
4    term        \rightarrow  factor mulop term
5    factor      \rightarrow  power
6    factor      \rightarrow  power powop factor
7    power       \rightarrow  ( expression )
8    power       \rightarrow  - power
9    power       \rightarrow  id
10   addop       \rightarrow  +
11   addop       \rightarrow  -
12   mulop       \rightarrow  *
13   mulop       \rightarrow  /
14   powop       \rightarrow  ^
```

Figure 18.2

a. \( p(+) \geq p(\text{unary-}) \geq p(*) \)
b. \( p(+) \geq p(-) \geq p(*) \geq p(\land) \)
c. \( p(\text{unary-}) \geq p(-) \geq p(*) \geq p(\land) \)
d. \( p(\land) \geq p(*) \geq p(\lor) \geq p(-) \)
19. (Ch. 7. p. 328) What is printed by the program given in the Figure below if operands on lines 9 and 11 are evaluated right to left?

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

int fun(int* k)
{
    *k += 4;
    return 3 * (*k) - 1;
}

int main()
{
    int i=5, j=7, sum1, sum2;
    sum1 = (i/2) + fun(&i);
    cout << " sum1= " << sum1;
    sum2 = fun(&j) + (j/2);
    cout << " sum2= " << sum2;
    cout << endl;
    return 0;
}
```

**Figure 19 Program for Question 19**

a. sum1=30 sum2=37  
b. sum1=28 sum2=37  
c. sum1=30 sum2=35  
d. sum1=28 sum2=35

20. (Ch. 8. p. 359) What programming language feature illustrated in the code fragment below is prohibited in C#?

```cpp
switch (value) {
    case -1:
        Negatives++;
        break;
    case 0:
        Zeros++;
        goto case 1;
    case 1:
        Positives++;
    default:
        Console.WriteLine("Default\n");
}
```

**Figure 20 Code Fragment for Question 20**

a. implicit execution of more than one selectable segment  
b. unary assignment operators ++ and --.  
c. goto statements  
d. negative case values.
21. (Ch. 8. p. 373) What programming language contains the grammar given below for a for-statement?

<table>
<thead>
<tr>
<th>for-statement</th>
<th>→</th>
<th>for loop-variable in object statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>for-statement</td>
<td>→</td>
<td>for loop-variable in object statement else statement</td>
</tr>
</tbody>
</table>

a. CLU.
b. Ada
c. Perl
d. Python

22. (Ch. 8. p. 368) What is the value of variable Count after the Ada code fragment given below exits the loop?

```ada
Count: Float:=1.35;
for Count in 1..10 loop
    Sum:=Sum+Count;
end loop;
```

Figure 22 Program for Question 22

a. 10.35
b. The code fragment fails to compile because Ada does not permit a floating point variable to be a loop control variable.
c. 1.35
d. 11

23. (Ch. 8. p. 354) In what language is the following example prohibited?

```plaintext
if (sum==0)
    if (count==0)
        result=0;
    else
        result=1;
```

Figure 23 Code Fragment for Question 23

a. Perl
b. Java
c. C#
d. C++
24. (Ch. 8. p. 353) What is printed when the program below is executed?

```cpp
#include <iostream>
using namespace std;
int main()
{
    int sum=0, result;
    if (sum=1) result=0; else result=1;
    cout << "sum=" << sum << " result=" << result;
    cout << endl;
    return 0;
}
```

**Figure 24 Program for Question 24**

a. sum=0 result=0  
b. sum=0 result=1  
c. sum=1 result=0  
d. sum=1 result=1

25. (Ch. 8. p. 383) What is the relationship between integer variables \( m1, m2, \) and \( m3 \) after exiting the loop containing guarded command proposed by Dijkstra shown below?

```cpp
do m1 > m2 -> t:=m1; m1:=m2; m2:=t;  
[] m2 > m3 -> t:=m2; m2:=m3; m3:=t;  
end
```

**Figure 25 Program for Question 25**

a. \( m1 \leq m2 \leq m3 \)  
b. \( m3 \leq m2 \leq m1 \)  
c. \( m1 < m2 < m3 \)  
d. \( m3 < m2 < m1 \)