University of Michigan
EECS 183: Elem. Programming Concepts
Fall 2011

Exam 1: Part 1: Form 1

Professors: ML Dorf, Elliot Soloway

Wed   9-February-2011

35 questions * 3 pts each = 105 pts
(yes we know there are 37 questions)

Instructions

- **Bubble in your name, UMID, AND FORM** on **scantron** sheet. 'section' is not needed.

- This course operates under the rules of the college of Engineering Honor Code. Your signature endorses the pledge below. After you finish your exam, **sign the scantron sheet to indicate you have adhered to the Honor Pledge**.

- The exam is **closed book**. No books, notes or the like may be used.

- This exam is **closed electronic devices**. No computers, calculators, PDAs, cell phones or other electronic devices may be used. I know texting is a good thing – just not during an exam.

- The exam is **closed neighbor**. No 'partner' on this -- only on projects -- NO talking with other students during the exam. NO sign languages. NO passing notes. NO communication of any type or form.

- Some questions are not simple, therefore, read carefully

- Assume all code and code fragments compile, unless otherwise specified.

- Assume/use only the standard ISO/ANSI C++.

- Assume "valid" means compiles and runs in standard ISO/ANSI C++

> ### Honor Pledge:
>
> "I have neither given nor received aid on this exam, nor have I concealed any violations of the Honor Code."

Basics

1. Which of these has the highest precedence?

   A. /
   B. %
   C. +
   D. *
   E. ()

2. Which of these has the lowest precedence priority?

   A. ||
   B. +
   C. *
   D. I don't know
   E. !

3. What prints? (Think carefully)
   ```
   double x = 1000 * 1000 * 1000 * 4 * 0.5;
   cout << x << endl;
   ```

   A. 2000000000
   B. 2000000000.0000
   C. 2E09
   D. cannot be determined
   E. multiple of the above are correct

4. What will be the result of `100/10*2`?

   A. Other
   B. I don't know
   C. 5
   D. 20

5. What will be the results of `1000 / 100 % 7 * 2`?

   A. Other
   B. 10
   C. 6
   D. 250
   E. 1000

6. What does this C++ expression `( 5 / 2 ) * 4` evaluate to?

   A. 0
   B. 2
   C. 8
   D. 10

7. What does this C++ expression `( 5 / 2 ) * 4.0` evaluate to?

    A. `0.0`
    B. `2.0`
    C. `8.0`
    D. `10.0`

8. Why is using named constants important in programming?

    A. `to create self-documenting code`
    B. `to make the program more maintainable`
    C. `to make debugging easier`
    D. `all of the above`
    E. none of the above

9. What do you need to `#include` to be able to use

```
int x = 0;

cout << x + 5 << endl;
```

    A. `#include <iostream>`
    B. `#include <cmath>`
    C. `#include <string>`
    D. `#include <iomanip>`
    E. `none of the above`

10. What do you need to `#include` to be able to use:

```
int x = 0;

x = x * 5 - 7.2 / 3;
```

    A. `#include <iostream>`
    B. `#include <cmath>`
    C. `#include <string>`
    D. `#include <iomanip>`
    E. `none of the above`

11. In C++, which of the following is NOT an operator

    A. !=
    B. ++
    C. cin
    D. /
    E. All the above are operators

12. Given a `bool` variable isFull, which of the following is a valid C++ assignment statement?

```
A. isFull = true;
B. isFull = isFull || ceil(2.4) % 2 + 1 == 3;
C. isFull = 2 < 3 + 2 < 20 - 5;
D. all of the above
E.  none of the above
```

13. Which of the following code fragments outputs the number 3 to the screen?

```
A. cout << 1 + 2 * 3 / 4 << endl;
B. cout << 1 * 2 + 3 / 4 + 2 << endl;
C. cout << 2 * (2 + 3) / 4 << endl;
D. cout << 1 / 2 * 2 + 3 << endl;
E. More than one of the above
```

14. Consider the following code fragment:

```
int x = 0, y = 0, z = 0;
char ch;
cin >> x >> ch
    >> y >> ch
    >> z >> ch;
cout << x + y + z;
```

If the user types

2.5 + 3 - 10

and presses return at the standard input, what does the above code fragment print?

```
A. 0
B. 7
C. 8
D. 10
E. None of the above
```

15. What is the output when the following code fragment is executed?

```
int    n;
double x = 3.8;
n = (int)x;
cout << n << endl;
```

```
A. 3
B. 4
C. 8
D. this will not compile
E. none of the above
```

16. Which of the following is a valid function prototype?

```
A. float some Function();
B. void nothing;
C. int (int thing);
D. void something();
```

17. Given the prototype: `void foo(int x);` which of the following calls are valid.
    Select all choices that apply.

    ```
    A. cout << foo(42);
    B. int y = foo(15);
    C. foo(-5);
    D. int y = 5 + foo(6);
    ```

18. Which of the following is equivalent to `( !(p < q) )`?

    ```
    A.  p < q
    B.  p >= q
    C.  p > q
    D.  !p >= ! q
    E.  none of the above
    ```

19. Can a **return** statement be used in a **void** function?

    ```
    A. Yes, and it will return where it was called from
    B. Yes, and the return statement will be ignored
    C. No
    D. Depends
    ```

20. What is the output of this line of code, given the following definition?
    ```
    cout << aNumber(2,5); // code in main

    int aNumber (int x, int y) // function definition
    {
        return (2 * x - y);
    }
    ```

    ```
    A. -6
    B. -1
    C. 6
    D. 8
    ```

21. What is the output of the following line of code, given the function definition that
   follows:

```
cout << calc (1,2) + calc(2,3); // code in main

int calc (int x, int y)              // function definition
{
   x = x + 1;
   return x % y;
}
```

   A. 0
   B. 1
   C. 2
   D. none of the above

22. Which of the following function calls is valid for the function prototype?

```
       void calculation (int m, char p, double q);
```

   A. cout << calculation (3, "y", 2.0);
   B. cout << calculation (10, 'r', 7);
   C. calculation (10, 'r', 7.5);
   D. two of the above
   E. none of the above

23. Consider the following function definition:

```
int truncate(double bigAndUgly)
{
   int smallAndPretty = static_cast<int>(bigAndUgly);
   return smallAndPretty;
}
```

Which of the following is/are valid calls to this function:

   A. double value = truncate(44.5);
   B. int value = truncate(5);
   C. truncate(5.0);
   D. all the above
   E. none of the above

24. What is the output of the following line of code, given the function definition that follows:

```
cout << calc (calc(2,3),2); // code in main

int calc (int x, int y)
{
    x = x + 1;
    return x % y;
}
```

A. 0
B. 1
C. 2
D. none of the above

25. What value gets printed by the program?

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

int foo(int x, int y);
int bar(int x, int y);

int main( )
{
  const int x = 3;
  const int y = 2;

  cout << bar(x,y) << endl;

  return 0;
}

int foo(int x, int y)
{
  return x+y;
}

int bar(int x, int y)
{
  return  foo(x, y) +1;
}
```

A. 3
B. 5
C. 6
D. Ill-formed – will not compile
E. Un-defined

26. Given the following code fragment, what prints? Yes we know there are no { }

```
int x = 5;

if ( ( x != 0 ) && ( 10 / x == 2 ) )
       cout << "HERE";
else
       cout << "THERE";
```

A. HERE
B. THERE
C. the above code causes a division by zero error
D. none of the above

27. Given the following code fragment, what prints? Yes we know there are no { }

```
int x = 0;

if ( ( x != 0 ) && ( 10 / x == 2 ) )
       cout << "HERE";
else
       cout << "THERE";
```

A. HERE
B. THERE
C. the above code causes a division by zero error
D. none of the above

28. What value gets printed by the program?

```
#include <iostream>

int main(  )
{
   int x = 0;
   int y = 0;

   if (x++ && y++)
   {
     y += 2;
   }

   cout << x + y << endl;

   return 0;
}
```

A. 1
B. 2
C. 3
D. 4
E. Undefined

The following code fragment applies to the next 2 questions:

What is the output of the following code:
```
if (age < 5)   {
     cout << "You will attend kindergarten soon." << endl;
} else if (age > 10)   {
     cout << "You are in elementary school." << endl;
} else if (age < 15)   {
     cout << "Congrats you are in middle school." << endl;
} else if (age < 18)   {
     cout << "Almost there ..." << endl;
} else  {
     cout << "Go BLUE! " << endl;
}
```

29. What is the output if the age variable is 15?

```
A. You will attend kindergarten soon.
B. You are in elementary school.
C. Congrats you are in middle school.
D. Almost there ...
E. GO BLUE!
```

30. What if the age variable is 10?

```
A. You will attend kindergarten soon.
B. You are in elementary school.
C. Congrats you are in middle school.
D. Almost there ...
E. GO BLUE!
```

31. If we have: (Yes we know there are no { })

```
int foo(int x)
{
   if(x>2)
      return  x  +  foo(x-1);
   else
      return 0;
}
```

What will be the result of: `cout << foo(3) << endl;`

```
A. Program will freeze
B. 6
C. 3
D. 10
E. 0
```

For the next 2 questions you are given: (Yes we know there are no { })

```
if (n < 10)
    if (n > 0)
        cout << "HERE" << endl;
 else
    cout << "THERE" << endl;
 cout << "I'm LOST" << endl;
 cout << "GO BLUE!" << endl;
```

32. What is the first line output if the variable n has the value 7 ?

    A. HERE
    B. THERE
    C. I'm LOST
    D. GO BLUE!

33. What is the first line output if the variable n has the value of 15?

    A. HERE
    B. THERE
    C. I'm LOST
    D. GO BLUE!

34. Consider the following function:

```
int difference(int x, int y)
{
   if ( x > y ) {
      return x - y;
   } else if ( x < y ) {
      return y - x;
   } else {
      return 0;
   }
}
```
Which of the following test suites thoroughly tests the above function by testing every path through the function?

    A. (1,-1), (2,-2), (3,-3)
    B. (0, 0), (0, 1), (0, 2)
    C. (1, 1), (1, 2), (2, 1)

35. Consider the following C++ function:
```
int max(int x, int y, int z)
{
   if ( x > y && x > z ) {
      return x;
   } else if ( y > x && y > z ) {
      return y;
   } else {
      return z;
   }
}
```

What does the C++ expression `max(1,2,3) + max(20,20,10)` evaluate to?

    A. 23
    B. 22
    C. 21
    D. none of the above

36. Consider the following code fragment:
```
int x = 7;
if ( x == 7 / 2 * 2 )
   cout << 'a';
   cout << 'b';
cout << 'c';
```

What does the above code print? Read carefully!

    A. abc
    B. bc
    C. c
    D. 'c'
    E. none of the above

37. Consider the following code fragment:
```
double x = 7;
if ( x = 7 / 2 * 2 )
   cout << 'A';
else
   cout << 'B';
```

What does the above code print? Read carefully!

    A. A
    B. B
    C. 'A'
    D. 'B'
    E. none of the above