

**CS 1044 Introduction to Programming in C      Test 2**

**Wednesday, December 4, 1996**

**Instructions:**

- Print your name in the space provided below.
- Print your name and ID number on the OpSCAN form.
- Be sure to code your ID number on the OpSCAN form as well and to enter Form **A** on the OpSCAN form.
- Choose the best answer for each question — some answers may be partially correct.
- Note that questions about printed values disregard formatting completely.
- When you have completed the test, sign the pledge at the bottom of this page and turn in both the test and your OpSCAN.
- **Note that failure to return this test or to discuss its content with a student who has not taken it is a violation of the Honor Code.**
- There are 40 questions — each is worth 2.5 points.
- I suggest you mark your answers on your test sheet as well as the OpSCAN form for your future reference. In any case, the answers you mark on the OpSCAN form will be considered your official answers.

<p><b>Do not start the test until instructed to do so!</b></p>
--

**Name** \_\_\_\_\_

<p><b>Pledge:</b> On my honor, I have neither given nor received unauthorized aid on this examination.</p> <p>_____</p>
---



**For questions 1 and 2, consider the program below:**

```
#include <iostream.h>
void main() {
    int count = 3, pass = 2;
    while (pass < 5) {
        pass = pass + 1;
        count = count * 2;
    }
    cout << pass << count << endl;
}
```

- 1) **What value is output for the variable `pass`?**  
a) 2                      b) 4                      c) 5                      d) 6                      e) none of the above
  - 2) **What value is output for the variable `count`?**  
a) 6                      b) 12                      c) 24                      d) 48                      e) none of the above
- 
- 3) **In a `while` loop, the condition that determines whether the loop is executed is viewed as being evaluated**  
a) at the end of the loop  
b) somewhere in the middle of the loop  
c) at the beginning of the loop  
d) not really done because, prior to executing the program the compiler calculates how many times the loop is to be executed  
e) none of the above
  - 4) **Which of the following are iteration mechanisms in C++?**  
a) `while`                      b) `switch`                      c) `for`  
d) `if ... else`                      e) all of the above                      f) a and c only  
g) a, c and d only                      h) none of the above
  - 5) **In a C++ program, whether a parameter is passed by value is indicated in the:**

- |                              |                               |
|------------------------------|-------------------------------|
| <b>a) function prototype</b> | <b>b) function definition</b> |
| <b>c) function calls</b>     | <b>d) all of the above</b>    |
| <b>e) a and b only</b>       | <b>f) b and c only</b>        |
| <b>g) a and c only</b>       | <b>h) none of the above</b>   |

**For questions 6 through 8, what value is output for the variable `anInt` by the indicated statement in the program given below?**

```
#include <iostream.h>
void func(int& x);
void main() {
    int anInt = 0;
    func(anInt);
    cout << anInt << endl;
    anInt = 1;
    func(anInt);
    cout << anInt << endl;
    func(anInt);
    cout << anInt << endl;
}

void func(int& x) {
    switch (x) {
        case 0: x = 5;
                break;
        case 1: x = 7;
        case 2:
        case 3: x = 11;
                break;
        default: x = 0;
    }
}
```

**6) the first cout statement**

- a) 0          b) 5          c) 7          d) 11          e) none of the above**

**7) the second cout statement**

- a) 0          b) 5          c) 7          d) 11          e) none of the above**

**8) the third cout statement**

- a) 0          b) 5          c) 7          d) 11          e) none of the above**
- 

**For questions 9 and 10, what value will be output for the indicated variable by the program below?**

```
#include <iostream.h>
void Remainder(int& Eins, int Zwei);
void main() {
    int First = 117, Second = 11;
    Remainder(First, Second);
    cout << First << Second;
}

void Remainder(int& Eins, int Zwei)
{
    Eins = Eins % Zwei;
    Zwei = 2*Zwei;
}
```

**9) First**

- a) 117          b) 11          c) 7          d) 2          e) none of the above**

**10) Second**

- a) 11          b) 22          c) 0          d) 7          e) none of the above**
-

**For questions 11 and 12, consider the program**

```
#include <iostream.h>
void main() {
    int SwapEm(int , int );
    int Temp = 0,
        First = 375,
        Second = 410;
    SwapEm(First, Second);
    cout << Temp << endl;
    cout << First << Second;
}

int SwapEm(int First, int Second)
{
    int Temp;
    Temp = First;
    First = Second;
    Second = Temp;
    return Temp;
}
```

- 11) What value is output for the variable Temp?**
- a) 0      b) 375      c) 410      d) A runtime error would occur.  
e) none of the above
- 12) What value is output for the variable First?**
- a) 0      b) 375      c) 410      d) A runtime error would occur.  
e) none of the above
- 

**For questions 13 through 15, consider the program**

```
#include <iostream.h>
int QuadFunc(float x);
float Temp = 0.0;
void main() {
    int x = 2.6;
    cout << x << endl;
    cout << QuadFunc(x) << endl;
    cout << Temp << endl;
}

int QuadFunc(float x)
{
    Temp = x*x + 2.0*x + 1.5;
    return Temp;
}
```

- 13) What value is output for the variable x?**
- a) 2.6      b) 2      c) 3      d) A runtime error would occur.  
e) A compile-time error would occur.  
f) None of the above.
- 14) What value is output by the second cout statement?**
- a) 13.46      b) 16.5      c) 9      d) A runtime error would occur.  
e) A compile-time error would occur.  
f) None of the above.
- 15) What value is output for the variable Temp?**
- a) 0.0      b) 13.46      c) 16.5      d) A runtime error would occur.  
e) A compile-time error would occur.  
f) None of the above.

- 
- 16) In C++, is it illegal to place the definition of one function inside the definition of another function?
- a) yes                      b) no                      c) none of the above
- 17) In C++, may variables that are declared inside `main()`, but not inside any other sub-block of `main()`, be accessed by every other function defined in the same file?
- a) yes                      b) no                      c) none of the above
- 

For questions 18 through 24 consider writing a program that contains the following prototype and declarations. Indicate whether the statement given in each question is allowed (syntactically legal, no compiler error) in C++. Be careful with this one – the question is about official ANSI C/C++ and assumes that your program does not do any operator overloading.

```
const int size = 5;
void DoSomething(int a[], int length);
int first[size], second[size], third[size];
```

Answers for questions 16 through 22 are:

- a) yes, syntax is legal  
b) no, syntax is illegal  
c) none of the above
- 18) `third = first;`
- 19) `third = first + second;`
- 20) `third[3] = first[3] + second[3];`
- 21) `third[0] = first[1] + second[2];`
- 22) `DoSomething(first, size);`
- 23) `DoSomething(first[3], size);`
- 24) `DoSomething(first, 2);`
-

**For questions 25 through 28 consider the following program and determine the values printed out at the indicated points in the program (Ignore issues of spacing.)**

```
#include <iostream.h>
void f1(int a[], int b[]);
void f2(int a[], int b[], int c[]);
int f3(int a[], int b[]);
void display(int a[]);
void main() {
    int a[3] = {3,2,1};
    int b[3] = {-4,0,2};
    int c[3] = {1,4,1};
    int d = 0;
    f1(c, b);
    display(c);
    f2(c, a, b);
    display(c);
    d = f3(a, b);
    cout << d << endl;
}

void f1(int a[],int b[]){
    for (int i = 0; i < 3; i++) {
        if (a[i] >= b[i])
            a[i] = b[i];
    }
}

void f2(int a[],int b[],int c[]){
    for (int i = 0; i < 3; i++) {
        a[i] = b[i] + c[i];
    }
}

int f3(int a [],int b[]){
    int val = 0;
    for (int i = 0; i < 3; i++) {
        val += a[i] * b[i];
    }
    return val;
}

void display(int a[]) {
    cout << "{" << a[0] << " ,";
    cout << a[1] << " ,";
    cout << a[2] << "}" << endl;
}
```

- 25) What's printed by the call to display() after the call to f1()?**
- a) {0,0,0}      b) {1,4,1}      c) {3,2,1}      d) {1,2,1}  
e) none of the above
- 26) What's printed by the call to display() after the call to f2()?**
- a) {0,0,0}      b) {1,4,1}      c) {-1,2,3}      d) {1,2,3}  
e) none of the above
- 27) What value is printed for the variable d after the call to f3()?**
- a) -12      b) 0      c) 2  
d) -10      e) none of the above
- 28) Which of the formal parameters for the function f2() should be passed by constant reference?**
- a) a[]      b) b[]      c) c[]  
d) all of the above      e) a[] and b[] only      f) a[] and c[] only  
g) b[] and c[] only      h) none of the above
-

**For questions 29 through 32 consider the program below:**

```
#include <iostream.h>
void MaxVal(int x, int y);
int  A    = 0,           // line 1
     B    = 1,           // line 2
     Big = 5;            // line 3
void main() {
    int A    = 2,           // line 4
        C    = 3,           // line 5
        Big = 7;            // line 6
    MaxVal(A, B);
    cout << Big << endl; // line 7
}
void MaxVal(int x, int y) {
    if (x >= y)
        Big = x;
    else
        Big = y;
}
```

- 29) The variable A is used in this program as a:**
- a) formal parameter                      b) actual parameter  
c) both                                      d) neither
- 30) File-scoped variables are declared in lines:**
- a) 1, 2 and 3                              b) 4, 5 and 6  
c) all of the above                      d) none of the above
- 31) What value is output in line 7?**
- a) 5                                      b) 3                                      c) 7                                      d) none of the above
- 32) The variable assigned a value by the function MaxVal() is declared in line:**
- a) 3                                      b) 6                                      c) neither                                      d) both
- 
- 33) The binary search algorithm**
- a) requires, on average, more comparisons than a linear search  
b) requires, on average, the same number of comparisons as a linear search  
c) works correctly only if the list being searched is sorted in order  
d) both a and c  
e) both b and c  
f) none of the above

**Questions 34 through ?? consider the selection sort algorithm covered in class. Consider the incomplete outline for a selection sort function below. The function should take an integer array and its size and sort the array entries into increasing order.**

```
void SortIt(int Arr[], int N) {                                // line 1
    int Begin, SmallSoFar, Look, Tmp;                          // line 2
    for (Begin = 0; _____; Begin++) {                      // line 3
        SmallSoFar = _____;                               // line 4
        for (Look = Begin + 1; _____; Look++) {           // line 5
            if (Arr[Look] < Arr[SmallSoFar])                     // line 6
                SmallSoFar = Look;                               // line 7
        }
        Tmp = Arr[Begin];                                       // line 8
        Arr[Begin] = _____;                               // line 9
        Arr[SmallSoFar] = _____;                           // line 10
    }
}
```

**34) The blank in line 3 should contain:**

- |                     |                             |                         |
|---------------------|-----------------------------|-------------------------|
| <b>a)</b> Begin < N | <b>b)</b> Begin <= N        | <b>c)</b> Begin < N - 1 |
| <b>d)</b> Begin > 0 | <b>e)</b> none of the above |                         |

**35) The blank in line 4 should contain:**

- |                 |                             |                     |
|-----------------|-----------------------------|---------------------|
| <b>a)</b> Begin | <b>b)</b> Begin + 1         | <b>c)</b> Begin - 1 |
| <b>d)</b> 0     | <b>e)</b> none of the above |                     |

**36) The blank in line 5 should contain:**

- |                        |                             |                        |
|------------------------|-----------------------------|------------------------|
| <b>a)</b> Look < N     | <b>b)</b> Look <= N         | <b>c)</b> Look < N - 1 |
| <b>d)</b> Look < Begin | <b>e)</b> none of the above |                        |

**37) The blank in line 9 should contain:**

- |                      |                             |                           |
|----------------------|-----------------------------|---------------------------|
| <b>a)</b> Arr[Begin] | <b>b)</b> Arr[Look]         | <b>c)</b> Arr[SmallSoFar] |
| <b>d)</b> Tmp        | <b>e)</b> none of the above |                           |

**38) The blank in line 10 should contain:**

- |                      |                             |                           |
|----------------------|-----------------------------|---------------------------|
| <b>a)</b> Arr[Begin] | <b>b)</b> Arr[Look]         | <b>c)</b> Arr[SmallSoFar] |
| <b>d)</b> Tmp        | <b>e)</b> none of the above |                           |
-

- 39) Which of the following describes what happens when an actual parameter is passed by reference?**
- a) the value of the actual parameter is copied into the formal parameter**
  - b) the formal parameter is treated as an alias or synonym for the actual parameter**
  - c) the formal parameter is treated as an alias or synonym for a temporary copy of the actual parameter**
  - d) all of the above**
  - e) none of the above**
- 40) Which of the following describes what happens when an actual parameter is passed by constant reference?**
- a) the value of the actual parameter is copied into the formal parameter**
  - b) the formal parameter is treated as an alias or synonym for the actual parameter**
  - c) the formal parameter is treated as an alias or synonym for a temporary copy of the actual parameter**
  - d) all of the above**
  - e) none of the above**