

Instructions: This homework assignment focuses primarily on C++ functions. The answers to the following questions can be determined from Chapters 3 through 7 of the lecture notes and Chapters 2 through 8 of the text. Assume any `#include` directives, variable declarations, etc, which are needed to make the given code syntactically correct.

Opscan forms will be passed out in class. Print your name and code your ID number on the opscan form. Mark Group 1 if you are in the 8TTh section and Group 2 if you are in the 10MWF section. Turn in your completed opscan at class on March 25th or 26th, or to the 1044 GTAs in McB 116/118 between 12:00 and 3:00 on Friday March 26th. Opscans will not be accepted late or at any other time.

1) When parameters are passed between the calling code and the called function, formal and actual parameters are matched by:

- | | |
|---------------------|--|
| 1) their data types | 2) their relative positions in the formal and actual parameter lists |
| 3) their names | 4) whether they are inputs to or outputs from the function |
| 5) none of these | |

2) A parameter should be passed by value if that parameter's data flow is:

- | | |
|---|----------------------------------|
| 1) one-way, into the function. | 2) one-way, out of the function. |
| 3) two-way, into and out of the function. | 4) 1 and 2 above |
| 5) 2 and 3 above | 6) none of these |

3) Given the function prototype and declarations:

```
float Fix(int& , float );  
int someInt = 10;  
float someFloat = 4.3;
```

which of the following function calls would be syntactically correct?

- | | |
|---|--|
| 1) <code>Fix(someInt, 6.85);</code> | 2) <code>someFloat = Fix(24, 6.85);</code> |
| 3) <code>someFloat = 0.3 * Fix(someInt, 6.85);</code> | 4) <code>Fix(someInt + 5, someFloat);</code> |
| 5) all of the above | 6) 1 and 3 above |
| 7) 2 and 4 above | 8) none of the above |

4) A function `SomeFunc` has two formal parameters, `alpha` and `beta`, of type `int`. The data flow for `alpha` is one-way, into the function. The data flow for `beta` is two-way, into and out of the function. What is the most appropriate function prototype for `SomeFunc`?

- 1) `void SomeFunc(int alpha, int beta);`
- 2) `void SomeFunc(int& alpha, int beta);`
- 3) `void SomeFunc(int alpha, int& beta);`
- 4) `void SomeFunc(int& alpha, int& beta);`
- 5) 1 and 2 above
- 6) 3 and 4 above
- 7) none of these

5) Given the function definition

```
void Twist( int  a, int& b ) {  
    int c;  
  
    c = a + 2;  
    a = a * 3;  
    b = c + a;  
}
```

what is the output of the following code fragment that invokes Twist?

```
int r = 1;  
int s = 2;  
int t = 3;  
Twist(t, s);  
cout << r << ' ' << s << ' ' << t << endl;
```

- | | |
|----------------------|-----------|
| 1) 1 2 3 | 2) 1 2 9 |
| 3) 1 14 3 | 4) 1 10 3 |
| 5) 5 14 3 | 6) 1 14 9 |
| 7) none of the above | |

6) Which of the following statements about value parameters is true?

- 1) The actual parameter is never modified by execution of the called function.
- 2) The formal parameter is never modified by execution of the called function.
- 3) The actual parameter must be a variable.
- 4) The actual parameter cannot have a Boolean value.
- 5) 2 and 3 above
- 6) none of these

7) Which of the following statements about reference parameters is true?

- 1) The actual parameter can be modified by execution of the called function.
- 2) The formal parameter can be modified by execution of the called function.
- 3) The actual parameter cannot be a variable.
- 4) The actual parameter cannot have an integer value.
- 5) 1 and 2 above
- 6) none of these

8) Which of the following statements about constant reference parameters is true?

- 1) The actual parameter can be modified by execution of the called function.
- 2) The formal parameter can be modified by execution of the called function.
- 3) The actual parameter cannot be a variable.
- 4) The actual parameter cannot have an integer value.
- 5) 1 and 2 above
- 6) none of these

9) Consider the function definition

```
void Demo( int intVal, float& floatVal ) {  
    intVal    = intVal * 2;  
    floatVal = float(intVal) + 3.5;  
}
```

What values are printed by the following code fragment?

```
int    myInt    = 20;  
float myFloat = 4.8;  
Demo(myInt, myFloat);  
cout << "myInt = " << myInt << " and myFloat = " << myFloat << endl;
```

- 1) myInt = 20 and myFloat = 43.5
- 2) myInt = 40 and myFloat = 4.8
- 3) myInt = 20 and myFloat = 4.8
- 4) myInt = 40 and myFloat = 43.5
- 5) none of the above

10) Consider the function definition

```
void Demo( int& intVal, float floatVal ) {  
    intVal    = intVal * 2;  
    floatVal = float(intVal) + 3.5;  
}
```

What values are printed by the following code fragment?

```
int    myInt    = 20;  
float myFloat = 4.8;  
Demo(myInt, myFloat);  
cout << "myInt = " << myInt << " and myFloat = " << myFloat << endl;
```

- 1) myInt = 20 and myFloat = 43.5
- 2) myInt = 40 and myFloat = 4.8
- 3) myInt = 20 and myFloat = 4.8
- 4) myInt = 40 and myFloat = 43.5
- 5) none of the above

11) If an ampersand (&) is not attached to the data type of a formal parameter, then the corresponding actual parameter can be:

- 1) a constant
- 2) a variable name
- 3) an arbitrary expression
- 4) 1 and 2 above
- 5) 1, 2, and 3 above
- 6) none of the above

12) For the function definition

```
void Func( int& gamma ) {  
    gamma = 245;  
}
```

which of the following comments best describes the direction of data flow for gamma?

- 1) one-way, into the function
- 2) one-way, out of the function
- 3) two-way, into and out of the function
- 4) none of the above

13) For the function definition

```
void Func( int gamma ) {  
    cout << 3 * gamma;  
}
```

which of the following comments best describes the direction of data flow for gamma?

- | | |
|--|---------------------------------|
| 1) one-way, into the function | 2) one-way, out of the function |
| 3) two-way, into and out of the function | 4) none of the above |

14) For the function definition

```
void Func( int& gamma ) {  
    gamma = 3 * gamma;  
}
```

which of the following comments describes the direction of data flow for gamma?

- | | |
|--|---------------------------------|
| 1) one-way, into the function | 2) one-way, out of the function |
| 3) two-way, into and out of the function | 4) none of the above |

15) For the function definition

```
void Func( int& alpha, int beta ) {  
    int delta;  
  
    delta = beta;  
    alpha = beta;  
    beta = 23;  
    cout << delta * beta;  
}
```

what is the function precondition?

- 1) // Pre: alpha is assigned a value before the function is called
- 2) // Pre: beta is assigned a value before the function is called
- 3) // Pre: delta is assigned a value before the function is called
- 4) // Pre: alpha and beta are assigned values before the function is called
- 5) // Pre: alpha, beta, and delta are assigned values before the function is called
- 6) none of the above

16) If a variable alpha is accessible only within function F, then alpha is either

- 1) a global variable or a formal parameter of F.
- 2) a local variable within F or a formal parameter of F.
- 3) a global variable or an actual parameter to F.
- 4) a local variable within F or an actual parameter to F.
- 5) none of the above

17) What is the output of the following code fragment?

```
int alpha = 3;
int beta = 20;
if (beta > 10) {
    int alpha = 5;

    beta = beta + alpha;
    cout << alpha << ' ' << beta << ' ';
}
cout << alpha << ' ' << beta;
```

- | | |
|--------------|----------------------|
| 1) 3 20 | 2) 3 25 3 25 |
| 3) 5 25 5 25 | 4) 5 25 3 25 |
| 5) 5 25 3 20 | 6) none of the above |

18) What is the appropriate function prototype for a function that receives a character letter grade and returns its integer equivalent on a four-point grading scale?

- | | |
|---------------------------|---------------------------|
| 1) void IntEquiv(char); | 2) void IntEquiv(int); |
| 3) int IntEquiv(char); | 4) int IntEquiv(char&); |
| 5) char IntEquiv(int); | 6) none of the above |

19) This question demonstrates the hazard of choosing inappropriate parameter-passing mechanisms. Given the function definition

```
int Power(int& base, int& exp ) {
    int product = 1;

    while (exp >= 1) {
        product = product * base;
        exp--;
    }
    return product;
}
```

what is the output of the following code?

```
int n = 2;
int pow = 3;
int result = Power(n, pow);
cout << n << " to the power " << pow << " is " << result;
```

- | | |
|--------------------------|--------------------------|
| 1) 2 to the power 3 is 8 | 2) 2 to the power 0 is 8 |
| 3) 0 to the power 0 is 0 | 4) 2 to the power 3 is 1 |
| 5) none of the above | |

20) What is the output of the following program?

```
#include <iostream.h>

void Try( int&, int );

int x, y, z;

void main( ) {
    x = 1;
    y = 2;
    z = 3;
    Try(y, x);
    cout << x << ' ' << y << ' ' << z << endl;
}

void Try( int& a, int b ) {
    int x;

    x = a + 2;
    a = a * 3;
    b = x + a;
}
```

- 1) 10 6 3
- 3) 1 2 3
- 5) none of the above

- 2) 10 2 3
- 4) 1 6 3