Alexander DuPree 2/20/2018 (Answers are in **BOLD)**

Week 7 - In-Class Worksheet

*1.*  Suppose

void nPrint(char ch, int n)   
{  
   while (n > 0)   
   {  
     cout << ch;  
     n--;  
   }  
}  
What is the printout of the call nPrint('a', 4)?

A. invalid call

B. aaaaa

**C. aaaa**

D. aaa

*2.*  The following code displays \_\_\_\_\_\_\_\_\_\_\_\_\_\_.  
  
  #include <iostream>  
  using namespace std;  
  
 void maxValue(int value1, int value2, int max);

  int main()  
  {  
    int max = 0;  
    maxValue(1, 2, max);  
    cout << "max is " << max << endl;  
  
    return 0;  
  }

  void maxValue(int value1, int value2, int max)  
  {  
    if (value1 > value2)  
      max = value1;  
    else  
      max = value2;  
  }

A. max is 2

**B. max is 0**

C. max is

D. max is 1

*3.*  Suppose   
  
void nPrint(char ch, int n)   
{  
   while (n > 0)   
   {  
     cout << ch;  
     n--;  
   }  
}  
  
What is k after invoking nPrint('a', k)?  
  
int k = 2;  
nPrint('a', k);

A. 0

B. 1

**C. 2**

D. 3

Show the output of the following programs.

*4*.

#include <iostream>

using namespace std;

void xFunction(int i);

int main()

{

int i = 1;

while (i <= 5)

{

xFunction(i);

i++;

}

return 0;

}

void xFunction(int i)

{

int num = 1;

for (int j = 1; j <= i; j++) {

cout << num << " ";

num \*= 3;

}

cout << endl;

}

***OUTPUT:*** 1

1 3

1 3 9

1 3 9 27

1 3 9 27 81

*5*.

#include <iostream>

using namespace std;

void xfunction(int i); **// lowercase ‘f’ used in prototype will throw an error**

int main()

{

int i = 1;

while (i <= 5)

{

xFunction(i);

i++;

}

cout << "i is " << i;

return 0;

}

void xFunction(int i)

{

do

{

if (i % 2 != 0)

cout << i << " ";

i--;

}

while (i >= 1);

cout << endl;

}

**OUTPUT :** 1

1

3 1

3 1

5 3 1

i is 6

*6*. Find the error in each of the following program segments and explain how the error can be corrected:

a.

int g()

{

cout << “Inside function g\n” << endl;

int h()

{

cout << “Inside function h\n” << endl;

return 0;

}

return 0;

}

**Declaring the function h() within the function g(). We need to prototype and define the function outside of g()’s scope.**

b.

int sum(int x, int y)

{

int result;

result = x + y;

}

**We need to return an int value here for this function to work. I.E. add: return result;**

c.

void sum(int n)

{

if (n==0)

return 0;

else

n = n + n;

}

**Technically this function will work. But we should be returning n in the else statement.**

d.

void f(float a);

{

float a;

cout << a << endl;

return;

}

**We declare the float ‘a’ again even though it was used as a parameter to f(). Get rid of the line: float a; and the function will work.**

*7*. Write a function header for each of the following functions:

1. Function **hypotenuse** that takes two double-precision floating point arguments, *side1* and *side2*, and returns a double-precision floating point result.

**double hypotenuse(double side1, double side2) {}**

1. Function **smallest** that takes three integers: *x, y, z* and returns an integer.

**int smallest(int x, int y, int z) {}**

1. Function **intToFloat** that takes an integer argument, *number*, and returns a floating point result.

**float intToFloat(int number) {}**

8. What is the difference between a parameter and an argument?

**A parameter is variable that is used in the function declaration. The argument is the actual value of the variable used when calling the function.**

9. Write a function for these functions:

1. Function **swap** that takes two integers by reference, *firstNum* and *secondNum*, and swaps the values in the two integers.
2. Function **largest** that takes three integers by reference, *num1, num2*  and *num3*, and returns the largest of the 3 numbers. (NOTE: The largest *does not use pass by reference.*)

**void swap(int &firstNum, int &secondNum)**

**{**

**int temp = firstNum;**

**firstNum = secondNum;**

**secondNum = temp;**

**return;**

**}**

**int largest(int &num1, int &num2, int &num3)**

**{**

**int largest = num1;**

**if (num2 >= largest) { largest = num2; }**

**if (num3 >= largest) { largest = num3; }**

**return largest;**

**}**

Week 7 - Pass by Reference - Lab Practice

1. Download the file, labPractice.cpp, from the class website.

2. Look over the program, making sure you understand what the program is supposed to do.

3. Write a function that is called twice, once to get values from the user for length and width of the lot and once to get values from the user for length and width of the structure. The function must use pass by reference.

4. Write a function that uses pass by reference to get values from the user for the length and width of the structure.

5. Write a function that uses pass by reference to compute the area of the yard.

6. Output should show all values: length & width of the lot & of the structure, and size of the yard. Pay attention to format of the output (of course).

7. Test and debug your program.

TEST DATA:

lot: 60 60 (measured in feet)

structure: 40 35 (measured in feet)

yard should end up being 2200 square feet