**Jeffrey Cho Chapter 6 Study Guide**

1. Using the functions described in Table 6-1, write each of the following as a C++ expression.

(The expression in (e) denotes the absolute value of x + 2y – 3.)

1. **double pow(double 8.50, double 3.2)**
2. **sqrt(3\*pow(x,2)-y)**
3. **(-b + sqrt(pow(b,2)-(4\*a\*c)))/(2\*a)**
4. **abs(x + (2\*y) – 3)**

2

. Consider the following function definition

:



int func(int x, double y, char u, string name)

{

//function body

}

Which of the following are correct function prototypes of the function func?

1. int func(x, y, u, name);
2. **int func(int s, double k, char ch, string name); \***
3. **int func(int, double, char, string); \***
4. func(int, double, char, string)

1. Consider the following statements:

int num1, num2, num3; double length, width, height; double volume;

num1 = 6; num2 = 7; num3 = 4; length = 6.2; width = 2.3; height = 3.4

and the function prototype:

double box(double, double, double);

Which of the following statements are valid? If they are invalid, explain why.

* 1. volume = box(length, width, height); **valid**
  2. volume = box(length, 3.8, height); **valid**
  3. cout << box(num1, num3, num2) << endl; **valid**
  4. cout << box(length, width, 7.0) << endl; **valid**
  5. volume = box(length, num1, height); **valid**
  6. cout << box(6.2, , height) << endl; **invalid; the second argument in the function call is missing.**
  7. volume = box(length + width, height); **invalid; the function call requires one more argument.**
  8. volume = box(num1, num2 + num3); **invalid; the function call requires one more argument.**

1. Short answer:
   1. Why do you need to include function prototypes in a program that contains user-defined functions?

**The function prototype gives the program the name of the function, the number and data types of the parameters, and the data type of the returned value: just enough information to let C++ use the function. It is also a promise that the full definition will appear later in the program. If you neglect to write the definition of the function, the program may compile, but it will not execute.**

* 1. Explain the difference between an actual and a formal parameter.

**A variable declared in the heading of a function definition is called a formal parameter. A variable or expression used in a function call is called an actual parameter.**

* 1. Explain the difference between a value and a reference parameter.

**A value parameter receives a copy of the actual parameter’s data. A reference parameter receives the address of the actual parameter.**

* 1. Explain the difference between a local and a global variable.

**A variable declared within a function or block is called a local variable. A variable declared outside of every function definition is called a global variable.**

1. Write the definition of a void function that takes as input a decimal number and outputs 3 times the value of the decimal number. Format your output to two decimal places.

**void f(double x)**

**{**

**cout << fixed << showpoint << setprecision(2);**

**cout << 3 \* x << endl;**

**}**

1. Write the definition of a void function with three reference parameters of type int, double, and string. The function sets the values of the int and double variables to 0 and the value of the string variable to an empty string.

**void f(int& x, double& y, string& s)**

**{**

**x = 0;**

**y = 0;**

**str = "";**

**}**

1. Write the definition of a void function that takes as input three parameters of type int, say sum, prod, and num. The function updates the value of sum by adding the value of num, and the value of prod by multiplying the value of num. The new values of sum and prod are reflected in the calling environment.

**Void f(int sum, int prod, int num)**

**{**

**Sum+=num;**

**Prod+=num;**

**Cout << “sum: “ << sum << “\nprod: “ << prod << endl;**

**}**

1. Write the definition of a function that takes as input the three numbers. The function returns true if the first number to the power of the second number equals the third number; otherwise, it returns false. (Assume that the three numbers are of type double.)

**Bool f(double x, double y, double z)**

**{**

**If (pow(x,y) == z)**

**Return true**

**Return false**

**}**

1. Consider the following functions:

int find(int num1, int num2)

{

int first, second; first = num1 \* num2; second = first - num1 + num2; if (second > 50) num1 = first / 10; else

num2 = second / 20; return num1 + num2;

}

int discover(int one, int two)

{

int secret = 1;

for (int i = one; i < two; i++) secret = secret \* i; return secret;

}

What is the output of each of the following program segments?

1. cout << find(15, 25) << endl; **62**
2. cout << discover(3, 9) << endl; **20160**
3. cout<<find(10,10)<<""<<discover(10,find(10,10))<<endl; **20213837312**
4. x = 20; y = 10;

cout << discover(x, y) << endl; **1**

10. What is the output of the following code fragment?

int alpha = 5; int beta = 10;

if (beta >= alpha)

{

int alpha; int temp; alpha = 20; temp = beta - alpha; beta = beta \* temp; cout << alpha << ' ' << beta << endl;

}

cout << alpha << ' ' << beta << endl;

**20 −100**

**5 −100**

11. Consider the following function prototype:

void funcDefaultParam(double x = 7.3, int y = 4, string z = "\*");

Which of the following function calls is correct?

1. **funcDefaultParam(); \***
2. **funcDefaultParam(2.8); \***
3. **funcDefaultParam(3.2, 0, "h"); \***
4. funcDefaultParam(9.2, "\*");
5. **funcDefaultParam(7, 3); \***