Activity #12: Value Returning Functions

Recorder's Report

Manager: Reader:

Recorder: Driver:

Date: Score: Satisfactory / Not Satisfactory

Record your team's answers to the key questions (marked with ) below.

a) Model 1, Question #4

b) Model 2, Question #9

c) Model 3, Question #11

Activity #12: Value Returning Functions

In this activity, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to value-returning functions in C++.

Content Learning Objectives

After completing this activity, students should be able to:

- Explain the meaning and purpose of a value-returning function
- Combine function calls with looping and branching statements
- Explain programs that use the same function multiple times
- Use tests for programs which include functions

Process Skill Goals

During the activity, students should make progress toward:

- Write code that includes function definitions and function calls
- Write programs using functions together with looping and branching statements

Model 1 A C++ Program

```
1  #include <iostream>
2
3  using namespace std;
4
5  int getSmaller(int num1, int num2) {
6    int smaller;
7    if (num1 < num2) {
8      smaller = num1;
9    } else {
10     smaller = num2;
11   }
12   return smaller;
13 }
14
15 int main() {
16   int userNum1, userNum2;
17   cout << "Enter two numbers separated by a space: ";
18   cin >> userNum1 >> userNum2;
19   int smallerNum = getSmaller(userNum1, userNum2);
20   cout << "The smaller number is " << smallerNum << endl;
21 }
22
```

Output:

```
Enter two numbers separated by a space: 52 36
The smaller number is 36
```

Refer to Model 1 above as your team develops consensus answers to the questions below.

Questions (10 min) Start time:

1. What does the function defined on lines 5-13 do? The code is found in `activity12a.cpp`.

2. So far you have created 'void' functions which do not send any information back to the calling code. A *value-returning* function does send information of a particular type back to the calling code.

a) What type of information does the function `getSmaller` return?

b) On what line of code is the function call to `getSmaller`?

c) On what line does the function `getSmaller` send back a value?

3. In a *void function* the *function call* is on a line by itself. Why is the function call in this model on the right-hand-side of an assignment statement?

4. If you wanted to alter this function to select and return the smaller of two `double` values, what changes would you need to make?

Model 2 A C++ Function

```
1 void getHypotenuse(double a, double b) {
2   double square = pow(a,2) + pow(b,2);
3   double squareRoot = sqrt(square);
4   cout << "The hypotenuse length is " << squareRoot << endl;
5 }
6
```

Refer to Model 2 above as your team develops consensus answers to the questions below.

Questions (20 min) **Start time:**

5. What does this function do?

6. Is this a *void function* or a *value-returning* function?

7. Suppose you wanted to turn this into a value-returning function that returns the hypotenuse length.

a) How would you change line 1 in the model?

b) What command would you add to the end?

c) What line would you remove and why?

8. What is the difference between using `cout` in a function and using `return` in a function?

9. Below is the framework for a program that drills students on addition problems. In particular, the program should do the following:

- Display five addition problems, one at a time, and let the student answer each
- Print the correct answer if the user enters an incorrect answer
- Print a congratulatory message if the student's answer is correct
- Keep track of the number of problems the student answers correctly
- Print a special message if the user gets all five problems right

Fill in the missing code to make this program function as described. The code is in `activity12b.cpp`.

```
1  // Print celebratory rocket
2
3  /* ANSWER A */ printRocket() {
4    cout << "Blast-off!" << endl;
5    cout << "  ^"    << endl;
6    cout << " /*\\"  << endl;
7    cout << "/***\\" << endl;
8    cout << "|***|"  << endl;
9    cout << "|***|"  << endl;
10   cout << "|#|#|"  << endl;
11   /* ANSWER B */
12 }
13
14 // give problem and check answer
15 bool giveProblem(/* ANSWER C */) {
16   int studentAns;
17   int correctAns = num1+num2;
18   cout << num1 << "+" << num2 << "=";
19   cin >> studentAns;
20   /* ANSWER D */
21 }
22
```

```
1  int main() {
2    srand(time(0));  // seed random num
3    int numCorrect = 0;
4    for(/* ANSWER E */) {
5      int num1 = rand() % 10 + 1;
6      int num2 = rand() % 10 + 1;
7      if (/* ANSWER F */) {
8        cout << "Correct!" << endl;
9        numCorrect++;
10     } else {
11       cout << "Incorrect! It is "
12            << (num1 + num2) << endl;
13     }
14   }
15   if( /* ANSWER G */ ) {
16     /* ANSWER H */
17   } else {
18     cout << "You got " << numCorrect
19          << " correct." << endl;
20   }
21 }
22
```

a) Line 3: The type of `printRocket`:

b) Line 11: A `return` statement:

c) Line 15: Parameters for `giveProblem`:

d) Line 20: A `return` statement:

e) Line 4: Setup for `for` loop:

f) Line 7: Function call in `if` condition:

g) Line 15: Condition for `if` statement:

h) Line 16: Command if all 5 correct:

Model 3 Program Output and the `main` Function

```
1  int main() {
2    int userNum,compNum;
3    char tryAgain;
4    srand(time(0));
5    do {
6      do {
7        cout << "Enter a number from 1 to 5: ";
8        cin >> userNum;
9        if (userNum < 1 || userNum > 5) {
10         cout << "Invalid Number!" << endl;
11       }
12     } while (userNum < 1 || userNum > 5);
13     compNum = rand() % 5 + 1;
14     cout << "Computer number: "
15          << compNum << endl;
16     cout << "Your number: "
17          << userNum << endl;
18     cout << getMessage(userNum,compNum);
19     cout << "Try again (y/n)? ";
20     cin >> tryAgain;
21   } while (tryAgain=='y');
22 }
23
```

Output:

```
Enter a number from 1 to 5:  7
Invalid Number!
Enter a number from 1 to 5:  3
Computer number:  3
Your number:  3
Congratulations!  You guessed
it.
Try again (y/n)?  y

Enter a number from 1 to 5:
3
Computer number:  4
Your number:  3
I'm sorry, your number is
smaller.
Try again (y/n)?  y

Enter a number from 1 to 5:
3
Computer number:  1
Your number:  3
I'm sorry, your number is
larger.
Try again (y/n)?  n
```

Refer to Model 3 above as your team develops consensus answers to the questions below.

Questions (20 min) Start time:

10. There is one user-defined function that is missing from this program.

a) What is the name of the missing function?

b) Is the function `void` or value-returning?

c) What are the type(s) of its parameters?

d) Write the header for the function.

11. Does the missing function contain any `cout` statements? Explain your reasoning. 

12. Suppose line 18 in the model was changed to just `getMessage(userNum,compNum);`. How would this change your answers to questions 10 and 11 above?

13. Now write the function definition assuming that the function header is as shown below. Test your code in the file `activity12c.cpp`.

```
string getMessage(int userNum, int compNum)
```

14. Write a different version of this same function assuming that line 18 in the model was changed to just `getMessage(userNum,compNum);` and using the following function header.

```
void getMessage(int userNum, int compNum)
```