CPT 111 – Principles of Programming
Week 10 Programming Lab
Functions II

Learning Outcomes:

- Understanding local and global variables
- Describe functions with parameters – passing by reference
- Demonstrate Overloading functions

---

1. What is the difference between a static local variable and a global variable?

2. What is the output of the following program?

```
#include <iostream>
using namespace std;

void myFunc(); // Function prototype

int main()
{
   int var = 100;
   cout << var << endl;
   myFunc();
   cout << var << endl;
   return 0;
}

// Definition of function myFunc
void myFunc()
{
   int var = 50;
   cout << var << endl;
}
```

3. What is the output of the following program? Explain your answer.

```
#include <iostream>
using namespace std;

void showVar(); // Function prototype

int main()
{
   for (int count = 0; count < 10; count++)
      showVar();
   return 0;
}

void showVar() // Definition of function showVar
{
   static int var = 10;
   cout << var << "\t";
   var++;
}
```

4. Modify Question 3 such that variable `var` is not declared as static variable. What is the output? Explain your answer.

```
void showVar() // Definition of function showVar
{
   int var = 10;
   cout << var << "\t";
   var++;
}
```

5. What is the output of the following program? Explain your answer.

```
#include <iostream>
using namespace std;

void showVal(int=5, int=10); // Function prototype with default arguments

int main()
{
   showVar(); //Function call
   showVar(9);
   showVar(9, 99);
   return 0;
}

void showVal() // Function definition
{
   cout<< x << "\t" << y <<endl;

}
```

6. Define the prototype and header for a function called `compute`. The function should have three parameters: an `int`, a `double`, and a `long` (not necessarily in that order). The `int` parameter should have a default argument of 5, and the `long` parameter should have a default argument of 65536. The `double` parameter should not have a default argument.

7. Define the prototype and header for a function called `calculate`. The function should have three parameters: an `int`, a reference to a `double`, and a `long` (not necessarily in that order.) Only the `int` parameter should have a default argument, which is 47.

8. What is the output of the following program?

```
#include <iostream>
using namespace std;

void test(int = 2, int = 4, int = 6);

int main()
{
   test();
   test(6);
   test(3, 9);
   test(1, 5, 7);
   return 0;
}

void test (int first, int second, int third)
{
   first += 3;
   second += 6;
   third += 9;
   cout << first << " " << second << " " << third << endl;
}
```

9. The following program asks the user to enter two numbers. What is the output of the program if the user enters 12 and 14? Explain the effects of using parameters passing by reference.

```
#include <iostream>
using namespace std;

void func1(int &, int &);
void func2(int &, int &, int &);
void func3(int, int, int);

int main()
{
   int x = 0, y = 0, z = 0;
   cout << x << " " << y << " " << z << endl;
   func1(x, y);
   cout << x << " " << y << " " << z << endl;
   func2(x, y, z);
   cout << x << " " << y << " " << z << endl;
   func3(x, y, z);
   cout << x << " " << y << " " << z << endl;
   return 0;
}

void func1(int &a, int &b)
{
   cout << "Enter two numbers: ";
   cin >> a >> b;
}

void func2(int &a, int &b, int &c)
{
   b++;
   c--;
   a = b + c;
}

void func3(int a, int b, int c)
{
   a = b - c;
}
```

10. Lowest Score Drop - Write a program that calculates the average of a group of test scores, where the lowest score in the group is dropped. It should use the following functions:

- `void getScore()` should ask the user for a test score, store it in a reference parameter variable, and validate it. This function should be called by `main` once for each of the five scores to be entered.
- `void calcAverage()` should calculate and display the average of the four highest scores. This function should be called just once by `main` and should be passed the five scores.
- `int findLowest()` should find and return the lowest of the five scores passed to it. It should be called by `calcAverage`, which uses the function to determine which of the five scores to drop.

Input Validation: Do not accept test scores lower than 0 or higher than 100.

11. Star Search - A particular talent competition has five judges, each of whom awards a score between 0 and 10 to each performer. Fractional scores, such as 8.3, are allowed. A performer's final score is determined by dropping the highest and lowest score received, then averaging the three remaining scores. Write a program that uses this method to calculate a contestant's score. It should include the following functions:

- `void getJudgeData()` should ask the user for a judge's score, store it in a reference parameter variable, and validate it. This function should be called by `main` once for each of the five judges.
- `void calcScore()` should calculate and display the average of the three scores that remain after dropping the highest and lowest scores the performer received. This function should be called just once by `main` and should be passed the five scores.

The last two functions, described below, should be called by `calcScore`, which uses the returned information to determine which of the scores to drop.

- `int findLowest()` should find and return the lowest of the five scores passed to it.
- `int findHighest()` should find and return the highest of the five scores passed to it.

Input Validation: Do not accept judge scores lower than 0 or higher than 10.

10

12. Days Out - Write a program that calculates the average number of days a company's employees are absent. The program should have the following functions:

- A function called by `main` that asks the user for the number of employees in the company. This value should be returned as an `int`. (The function accepts no arguments.)
- A function called by `main` that accepts one argument: the number of employees in the company. The function should ask the user to enter the number of days each employee missed during the past year. The total of these days should be returned as an `int`.
- A function called by `main` that takes two arguments: the number of employees in the company and the total number of days absent for all employees during the year. The function should return, as a `double`, the average number of days absent. (This function does not perform screen output and does not ask the user for input.)

Input Validation: Do not accept a number less than 1 for the number of employees. Do not accept a negative number for the days any employee missed.

11

13. Overloaded Hospital - Write a program that computes and displays the charges for a patient's hospital stay. First, the program should ask if the patient was admitted as an in-patient or an out- patient. If the patient was an in-patient, the following data should be entered:

- The number of days spent in the hospital
- The daily rate
- Hospital medication charges
- Charges for hospital services (lab tests, etc.)

The program should ask for the following data if the patient was an out-patient:

- Charges for hospital services (lab tests, etc.)
- Hospital medication charges

The program should use two overloaded functions to calculate the total charges. One of the functions should accept arguments for the in-patient data, while the other function accepts arguments for out-patient information. Both functions should return the total charges.

Input Validation: Do not accept negative numbers for any data.

14. Paint Job Estimator - A painting company has determined that for every 110 square feet of wall space, one gallon of paint and eight hours of labor will be required. The company charges $25.00 per hour for labor. Write a modular program that allows the user to enter the number of rooms that are to be painted and the price of the paint per gallon. It should also ask for the square feet of wall space in each room. It should then display the following data:

- The number of gallons of paint required
- The hours of labor required
- The cost of the paint
- The labor charges
- The total cost of the paint job

Input validation: Do not accept a value less than 1 for the number of rooms. Do not accept a value less than $10.00 for the price of paint. Do not accept a negative value for square footage of wall space.

12

15. `isPrime` Function - A prime number is a number that is only evenly divisible by itself and 1. For example, the number 5 is prime because it can only be evenly divided by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6.

Describe a function name `isPrime`, which takes an integer as an argument and returns `true` if the argument is a prime number, or `false` otherwise. Demonstrate the function in a complete program.