Name:	Griffin Gowdey

Programming Fundamentals – Video Lesson

Functions

What is the advantage of breaking your application's code into several small functions?

> **Ans:**
>
> Each function can handle one small, manageable task. This makes it easier to design, code, test, and debug.

What is the difference between an argument and a parameter variable?

> **Ans:**
>
> Arguments appear in the parentheses of a function call. They are the actual values passed to a function. Parameters appear in the parentheses of a function heading. They are the variables that receive the arguments.

When a function accepts multiple arguments, does it matter what order the arguments are passed in?

> **Ans:**
>
> Yes. Example: pow() function that accepts the base value and exponent value in that order. The result of the pow function will change if the arguments are not passed in that order.

What does it mean to overload a function?

> **Ans:**
>
> Function overloading means including more than one function in the same program that has the same name. C++ allows the providing the overloaded functions can be distinguished by having different parameter lists.

If you are writing a function that accepts an argument and you want to make sure the function cannot change the value of the argument, what should you do?

> **Ans:**
>
> Pass it by value.

Give an example where an argument should be passed by reference.

> **Ans:**
>
> You want the function to change the value of a variable that is defined in the calling function.

How do you return a value from a function?

Ans:

> **With a return statement.**
> **return value;**
>
> **bool isMin(double a, double b)**
> **{**
> > **bool c = a < b;**
> > **return c;**
>
> **}**

Can a function have a local variable with the same name as a global variable?

Ans:

> **Yes, but within that function only that variable can be "seen" and accessed.**

When should a static local variable be used?

Ans:

> **Use a static variable when you need a local variable to retain its value between function calls.**

The following statement calls a function named half, which returns a value that is half that of the argument passed to it. Assume that result and number have both been defined to be double variables. Write the half function.

Ans:

> **double half(double value)**
> **{**
> > **return value / 2;**
>
> **}**

A program contains the following function.

Ans:

> **Int cube(int num)**
> **{**
> > **return num * num * num;**
> > **}**

Write a statement that passes the value 4 to this function and assigns its return value to the variable result.

Ans:

> **result = cube(4);**

Write a function, named timesTen, that accepts an integer argument. When the function is called, it should display the product of its argument multiplied times 10.

Ans:

```
void timesTen(int num)
{
    cout << num *10;
}
```

A program contains the following function.

Ans:

```
void display(int argl, double arg2, char arg3)
{
    cout << "Here are the values:" << argl << "";
    cout << arg2 << ""<< arg3 <<endl;
}
```

Write a statement that calls the function and passes the following variables to it:

int age;
double income;
char initial;

```
display(age, income, initial);
```

Write a function named getNumber, which uses a reference parameter to accept an integer argument. The function should prompt the user to enter a number in the range of 1 through 100. The input should be validated and stored in the parameter variable. 38.

Ans:

```
void getNumber (int & number)
{
    cout << "Enter an integer between 1 and 100): ";
    cin >> number;
    while (number < 1 || number > 100)
    {
        cout << "This value is out of the allowed range.\n"
        cout << "Enter an integer between 1 and 100): ";
        cin >> number;
    }
}
```

Write a function named biggest that receives three integer arguments and returns the largest of the three values

Ans:

```
int biggest(int num 1, int num2, int num3)
{
        if (num1 >= num2 && num1 >= num3)
                return num 1;
        if (num2 >= num3)
        return num2;
        return num3;
}
```

What is wrong with this code block?

```
void total(int value1, value2, value3) {
    total =  value1 + value2 + value3
    return total;
}
```

Ans:

Void functions do not return anything (value, expression, or variable). Each parameter must have a data type.

What is wrong with this code block?

```
double average(int value1, int value2, int value3) {
    double average;
    average = value1 + value2 + value3 / 3;
}
```

Ans:

The assignment statement should read:
average = (value1 + value2 + value3) / 3.0;
The function is declared as a double but returns no value.

What is wrong with this code block?

C)   ```
void area(int length = 30, int width)
{
        return length * width;
}
```

Ans:

(1) Change void to int
(2) width needs a default value too

```
int area(int length = 30, int width = 20)
{
        return length * width;
}
```

What is wrong with this code block?

D)   **void getValue(int value&)**
        {
                cout << "Enter a value:
                cin >> value&;
        }
        Ans:

     int value& should be
      int &value
     cin >> value&; should be
      cin >> value;

What is wrong with this code block?

E)   **// Overloaded functions**
        int getValue()
        {
                int inputValue;
                cout << "Enter an integer: ";
                cin >> inputValue;
                return inputValue;
        }

        double getValue() {
                double inputValue;
                cout << "Enter a floating-point number: ";
                cin >> inputValue;
                return inputValue;
                }
        Ans:

     Overload functions
      Same name
     Same of diff return type
      Parameter lists must be different
     Prototype
      double getValue(double num = 2.2);