**CS 131 Exercises**

**C#—A Beginner’s Guide**

**Pat McGee**

ISBN: 978-0-07-183581-7

**CS 131 Exercises - Chapter 3 : Numeric Operators and Routines, Random**

**Exercise 1.** Create a new C# Console project called Expressions.

* Declare an integer variable named ‘a’ initialized to the value 100, and another integer variable named ‘b’ initialized to the value 20.
* Output the result of the expression “a plus b” using the following code example:

Console.WriteLine("{0} + {1} = {2}", a, b, a + b);

* Create four more WriteLine statements using the following numeric operators: subtract, multiply, divide, remainder. Change the operator in both the string and the expression.

In this exercise, we created a project called expression which is intended to calculate using basic mathematical operations – addition; subtraction; multiplication; division; and remainder.

A close up of a logo

Description automatically generated

Figure 1.1: Output of Expression.cs

**Exercise 2.** Create a new C# Console project called BinaryOperators.

* Declare an integer variable named ‘a’ initialized to the value 100, and another integer variable named ‘b’ initialized to the value 20.
* Output the value of ‘a’ following the expression “assign ‘a plus b’ to a” using the following code example:

a += b;

Console.WriteLine("The value of a is now:{0}", a);

* Create four more binary operator statements that modify variable ‘a’ using the following numeric operators: subtract, multiply, divide, remainder.
* Repeat the WriteLine statement after each change to display the new value of variable ‘a’.

In this exercise, we created a project called BinaryOperators. This project is intended to calculate basic mathematical operations using compounding operators.

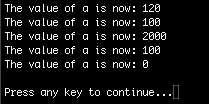


Figure 2.1: Output of BinaryOperators.cs

A screenshot of a cell phone

Description automatically generated

Figure 2.2: Reassigned value of `a` after addition

A screenshot of a cell phone

Description automatically generated

Figure 2.3: Reassigned value of `a` after addition

A screenshot of a cell phone

Description automatically generated

Figure 2.4: Reassigned value of `a` after subtraction

A screenshot of text

Description automatically generated

Figure 2.5: Reassigned value of `a` after subtraction

A screenshot of a cell phone

Description automatically generated

Figure 2.6: Reassigned value of `a` after multiplication

A screenshot of a cell phone

Description automatically generated

Figure 2.7: Reassigned value of `a` after multiplication

A screenshot of a cell phone screen with text

Description automatically generated

Figure 2.8: Reassigned value of `a` after division

A screenshot of a cell phone

Description automatically generated

Figure 2.9: Reassigned value of `a` after division

A screen shot of a computer

Description automatically generated

Figure 2.10: Reassigned value of `a` after taking the remainder

**Exercise 3.** Create a new C# Console project called Random.

* Repeat the steps in Exercise 1, replacing the hard coded values 20 and 100 with two randomly generated values between 1 and 100.
* Run the program multiple times and verify the variable values are randomly generated.

In this exercise, we are to replace the constant values initiated in Expressions.cs with system generated random number. Each run of the program should result to different output each time.

A close up of a logo

Description automatically generated

Figure 3.1: First run output of Random.cs

A close up of a logo

Description automatically generated

Figure 3.2: Second run output of Random.cs

A screenshot of a cell phone

Description automatically generated

Figure 3.3: Third run output of Random.cs