Purpose: Practice the use of lambdas, delegate and built-in delegate predicates/templates – Func<> and Action<>

Exercise 01: *[10 marks]*

Create a console C# app in which you are required to create the following methods:

a) string **Minimum**(string1, string2, string3) which returns the smallest of three string values. *[5 marks]*

To test this method, you need to use built-in **Func<>** delegate predicate/template and call the method using Func<> variable.

b) void **AvgGrade**(value1, value2, value3) which displays the average of three grades. *[5 marks]*

To test this method, you need to use built-in **Action**<> delegate predicate and call the method using Action variable.

Exercise 02:

a) Build a console app. Define a student's grade array having at least 10 double values between 5 and 50. Array must be populated with random values. You need to use loop and fill up the array with random values between 5 and 50. *[5 marks]*

b) Define a delegate – **GradesPredicate** which takes one input of type double and returns a bool. *[2 marks]*

c) Define a method – **GradesFilter** which displays only those grades values which are greater than or equal to 15. This method should take an array of type doubles and second argument a variable of GradesPredicate. This method should not return any type. We should call this method like this: **GradesFilter(gradesArray, lambda-Expression).** *Basically, we need to call this method with a lambda expression as second parameter.* *[10 marks]*

d) Lambda-Expression should filter values which are greater than or equal to 15. *[3 marks]*