using System;

using System.Reflection;

using System.Reflection.Emit;

namespace CalculatorIce

{

public class Program

{

public static void Main(string[] args)

{

//define the dynamic assembly name

AssemblyName calcAssemblyName = new AssemblyName("CalculatorAssembly");

//create the dynamic assembly and module

AssemblyBuilder assemblyBuilder = AssemblyBuilder.DefineDynamicAssembly(calcAssemblyName, AssemblyBuilderAccess.Run);

ModuleBuilder moduleBuilder = assemblyBuilder.DefineDynamicModule("CalcModule");

//define the Calculator type in the dynamic assembly

TypeBuilder typeBuilder = moduleBuilder.DefineType("Calculator", TypeAttributes.Public);

//define the Add method

MethodBuilder addMethodBuilder = typeBuilder.DefineMethod("Add", MethodAttributes.Public, typeof(int), new Type[] { typeof(int), typeof(int) });

ILGenerator addIL = addMethodBuilder.GetILGenerator();

addIL.Emit(OpCodes.Ldarg\_1);

addIL.Emit(OpCodes.Ldarg\_2);

addIL.Emit(OpCodes.Add);

addIL.Emit(OpCodes.Ret);

//define the Multiply method

MethodBuilder multiplyMethodBuilder = typeBuilder.DefineMethod("Multiply", MethodAttributes.Public, typeof(int), new Type[] { typeof(int), typeof(int) });

ILGenerator multiplyIL = multiplyMethodBuilder.GetILGenerator();

multiplyIL.Emit(OpCodes.Ldarg\_1);

multiplyIL.Emit(OpCodes.Ldarg\_2);

multiplyIL.Emit(OpCodes.Mul);

multiplyIL.Emit(OpCodes.Ret);

//create the Calculator type in the dynamic assembly

Type calculatorType = typeBuilder.CreateType();

//create an instance of the Calculator class

object calculatorInstance = Activator.CreateInstance(calculatorType);

//get user input for the integers

Console.Write("Enter the first integer: ");

int firstNumber = int.Parse(Console.ReadLine());

Console.Write("Enter the second integer: ");

int secondNumber = int.Parse(Console.ReadLine());

//invoke the Add method using reflection

MethodInfo addMethod = calculatorType.GetMethod("Add");

int addResult = (int)addMethod.Invoke(calculatorInstance, new object[] { firstNumber, secondNumber });

Console.WriteLine($"Add({firstNumber}, {secondNumber}) = {addResult}");

//invoke the Multiply method using reflection

MethodInfo multiplyMethod = calculatorType.GetMethod("Multiply");

int multiplyResult = (int)multiplyMethod.Invoke(calculatorInstance, new object[] { firstNumber, secondNumber });

Console.WriteLine($"Multiply({firstNumber}, {secondNumber}) = {multiplyResult}");

}

}

}