Curso C# Fundamentals for Absolute Beginners

Laboratorio de Desarrollo de Software

Daniel Germán - 1128127

8 de Enero, 2025

1.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World");
            Console.ReadLine();
        }
    }
}
```

2.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Variables
{
    class Program
    {
        static void Main(string[] args)
        {
            /*
            int x;
            int y;

            x = 7;
            y = x + 3;
```

```
            Console.WriteLine(y);
            Console.ReadLine();
            */

            Console.WriteLine("What's your name? ");
            Console.Write("Type your first name: ");
            string firstName;
            firstName = Console.ReadLine();

            /*
            Console.Write("What's your last name? ");
            Console.Write("Type your last name: ");
            string lastName;
            lastName = Console.ReadLine();
            */

            Console.Write("Type your last name: ");
            string myLastName = Console.ReadLine();

            Console.WriteLine("Hello, " + firstName + " " + myLastName);
            Console.ReadLine();

        }
    }
}
```

3.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Decisions
{
    class Program
    {
```

```
        static void Main(string[] args)
        {
            /*
            Console.WriteLine("Daniel's Big Giveaway");
            Console.Write("Choose door: 1, 2 or 3: ");
            string value = Console.ReadLine();

            string msg = "";

            if (value == "1")
                msg = "you have won a car";
            else if (value == "2")
                msg = "you won a new boat!";
            else if (value == "3")
                msg = "you won a new cat!";
            else
            {
                msg = "sorry, I didn't understand. ";
                //msg = msg + "You Lose.";
                msg += "You Lose.";
            }

            Console.WriteLine(msg);
            Console.ReadLine();
            */

            Console.WriteLine("Daniel's Big Giveaway");
            Console.Write("Choose a door: 1, 2 or 3: ");
            string userValue = Console.ReadLine();

            //if-else syntax = if value equals 1 then 'boat' else 'strand
of lint'
            string msg = (userValue == "1") ? "boat" : "strand of lint";

            //Console.Write("You won a ");
            //Console.Write(msg);
            //Console.Write(".");
            //Console.ReadLine();

            //string manipulation {} value replaced by msg
```

```
            //Console.WriteLine("You won a {0}.", msg);
            //Console.ReadLine();

            Console.WriteLine("You entered: {0}, therefore you won a
{1}.", userValue, msg);
            Console.ReadLine();

        }
    }
}
```

4.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OperatorsExpressionsStatements
{
    class Program
    {
        static void Main(string[] args)
        {
            //variable declaration
            int x, y, a, b;

            //assignment operator
            x = 3;
            y = 2;
            a = 1;
            b = 0;


            //addition operator
            x = 3 + 4;

            //subtraction operator
```

```
x = 10 - 5;

//multiplication operator
x = 3 * 5;

//division operator
x = 10 / 5;

// order of operations using paranthesis
x = (x + y) * (a - b);

//equality operator
if (x == y)
{
}

//greator than operatoer
if (x > y)
{
}

//less than operator
if (x < y)
{
}

//contional AND operator
if ((x > y) && (a > b))
{
}

//contional OR operator
if ((x> y) || (a > b))
{
}

string message = (x == 1) ? "car" : "Boat";

//Member access and Method invocation
Console.WriteLine("hi");
```

```
        }
    }
}
```

5.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ForIteration
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 0; i < 10; i++)
            {
              //  Console.WriteLine(i);
              if(i == 7)
                {
                    Console.WriteLine("Found seven!");
                    break;
                }
            }

            for (int myValue = 0; myValue < 12; myValue++)
            {
                Console.WriteLine(myValue);
            }

            Console.ReadLine();
        }
    }
}
```

6.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace UnderstandingArrays
{
    class Program
    {
        static void Main(string[] args)
        {
            /*
            int number1 = 4;
            int number2 = 8;
            int number3 = 15;
            int number4 = 16;
            int number5 = 23;

            if (number1 == 16)
            {
            }
            else if (number2 == 16)
            {
            }
            else if (number3 == 16)
            {
            }
            */

            /*
            //array of 5 indeces
            int[] numbers = new int[5];
            numbers[0] = 4;
            numbers[1] = 8;
            numbers[2] = 15;
```

```
            numbers[3] = 16;
            numbers[4] = 23;
            //numbers[5] = 42;

            Console.WriteLine(numbers[1]);
            Console.WriteLine(numbers.Length);
            Console.ReadLine();
            */


            //int[] numbers = new int[] { 4, 8, 15, 16, 23, 42 };
            //string[] names = new string[] { "Michael", "Robert",
"Alexander", "Frank" };

            /*
            for (int i = 0; i < names.Length; i++)
            {
                Console.WriteLine(names[i]);
            }
            Console.ReadLine();
            */

            /*
            foreach(string name in names)
            {
                Console.WriteLine(name);
            }
            Console.ReadLine();
            */

            string zig = "You can get what you want out of life if you
help enough other people get what they want.";

            char[] charArray = zig.ToCharArray();
            Array.Reverse(charArray);

            foreach(char zigchar in charArray)
            {
                Console.Write(zigchar);
            }
```

```
            Console.ReadLine();
        }
    }
}
```

7.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _7_HelperMethods
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

8.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WhileIteration
{
    class Program
    {
        static void Main(string[] args)
```

```
        {
            bool displayMenu = true;
            while (displayMenu)
            {
                displayMenu = MainMenu();
            }
        }

        private static bool MainMenu()
        {
            Console.Clear();
            Console.WriteLine("Choose an option: ");
            Console.WriteLine("1) Print Number");
            Console.WriteLine("2) Guessing Game");
            Console.WriteLine("3) Exit");

            string result = Console.ReadLine();
            if (result == "1")
            {
                PrintNumber();
                return true;
            }

            else if (result == "2")
            {
                GuessingGame();
                return true;
            }

            else if (result == "3")
            {
                return false;
            }
            else
            {
                return true;
            }
        }

        private static void PrintNumber()
```

```
        {
            Console.Clear();
            Console.WriteLine("Print number!");
            Console.WriteLine("Type a number: ");
            int result = int.Parse(Console.ReadLine());
            int counter = 1;
            while (counter < result + 1)
            {
                Console.Write(counter);
                Console.Write(" - ");
                counter++;
            }
            Console.ReadLine();
        }

        private static void GuessingGame()
        {
            Console.Clear();
            Console.WriteLine("Guessing Game");

            Random myRandom = new Random();
            int randonNumber = myRandom.Next(1, 11);

            int guesses = 0;
            bool incorrect = true;

            do
            {
                Console.WriteLine("Guess a number between 1 and 10");
                string result = Console.ReadLine();
                guesses++;
                if (result == randonNumber.ToString())
                    incorrect = false;
                else
                    Console.WriteLine("Wrong!");

            } while (incorrect);
            Console.WriteLine("Correct! it took you {0} guesses.",
guesses);
```

```
            Console.ReadLine();
        }
    }
}
```

9.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WorkingWithStrings
{
    class Program
    {
        static void Main(string[] args)
        {
            //string myString = "My \"so-called\" life";
//quotes
            //string myString = "What if I needa]nnew line?";      //new
line
            //string myString = "Go to your C:\\ drive";
//backslsh
            //string myString = @"Go to your C:\ drive";
//blackslash 2

            //string myString = String.Format("{1} = {0}", "First",
"Second");

            //string myString = String.Format("{0:C}", 1234);
            //string myString = String.Format( "{0:N}", 123456789);
            //string myString = String.Format("Percentage: {0:P}", .1234);
            //string myString = String.Format("Phone Number:
{0:(###)-(###)-(####)", 1234567890);
```

```
            //string myString = ("That summer we took three across the
bard ");

            ////myString = myString.Substring(6, 14);
            //myString = myString.ToUpper();
            //myString = myString.Replace(" ", "--");
            //myString = myString.Remove(6, 14);

            /*
            myString = String.Format("Length before : {0} -- length after:
{1}",
                myString.Length, myString.Trim().Length);
             */

            StringBuilder myString = new StringBuilder();
            for (int i = 0; i < 100; i++)
            {
                myString.Append("--");
                myString.Append(i);
            }

            Console.WriteLine(myString);
            Console.ReadLine();
        }
    }
}
```

10.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DatesAndTimes
{
    class Program
```

```
    {
        static void Main(string[] args)
        {
            DateTime myValue = DateTime.Now;
            //Console.WriteLine(myValue.ToString());
            //Console.WriteLine(myValue.ToShortDateString());
            //Console.WriteLine(myValue.ToShortTimeString());
            //Console.WriteLine(myValue.ToLongDateString());
            //Console.WriteLine(myValue.ToLongTimeString());

            //Console.WriteLine(myValue.AddDays(3).ToLongDateString());
            //Console.WriteLine(myValue.AddHours(3).ToLongTimeString());
            //Console.WriteLine(myValue.AddDays(-3).ToLongDateString());

            //Console.WriteLine(myValue.Month);

            //DateTime myBirthday = new DateTime(2006, 9, 11);
            //Console.WriteLine(myBirthday.ToShortDateString());

            DateTime myBirthday = DateTime.Parse("9/11/2006");
            TimeSpan myAge = DateTime.Now.Subtract(myBirthday);
            Console.WriteLine(myAge.TotalDays);

            Console.ReadLine();
        }
    }
}
```

11.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SimpleClasses
{
    class Program
```

```
    {
        static void Main(string[] args)
        {
            Car myCar = new Car();
            myCar.Make = "Oldsmobile";
            myCar.Model = "Cutlas supreme";
            myCar.Year = 1986;
            myCar.color = "Silver";

            Console.WriteLine("{0} {1} {2} {3}", myCar.Make, myCar.Model,
myCar.Year, myCar.color);

            decimal value = DetermineMarketValue(myCar);
            //Console.WriteLine("{0:C}", value);

            Console.WriteLine("{0:C}", myCar.DetermineMarketValue());

            Console.ReadLine();
        }

        private static decimal DetermineMarketValue(Car car)
        {
            decimal carValue = 100.0M;

            return carValue;
        }

        class Car
        {
            public string Make { get; set; }
            public string Model { get; set; }
            public int Year { get; set; }
            public string color { get; set; }

            public decimal DetermineMarketValue()
            {
                decimal carValue;

                if (Year > 1990)
                {
```

```
                    carValue = 10000;
                }
                else
                    carValue = 2000;

                return carValue;
            }
        }
    }
}
```

12.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

Namespace ObjectLifeTime
{
    class Program
    {
        static void Main(string[] args)
        {
            Car myCar = new Car();

            Car.MyMethod();

            /*
            myCar.Make = "Oldmobile";
            myCar.Model = "Cutlas Supreme";
            myCar.Year = 1986;
            myCar.Color = "Silver";
            */

            //overloaded constructor used with parameters
            //Car myThirdCar = new Car("Ford", "Escape", 2005, "White");
```

```
        /*
        Car myOtherCar;
        myOtherCar = myCar;

        Console.WriteLine("{0} {1} {2} {3}",
            myOtherCar.Make,
            myOtherCar.Model,
            myOtherCar.Year,
            myOtherCar.Color);

        myOtherCar.Model = "98";

        Console.WriteLine("{0} {1} {2} {3}",
            myCar.Make,
            myCar.Model,
            myCar.Year,
            myCar.Color);

        myOtherCar = null;


        Console.WriteLine("{0} {1} {2} {3}",
            myOtherCar.Make,
            myOtherCar.Model,
            myOtherCar.Year,
            myOtherCar.Color);

        myCar = null;
        */


        Console.ReadLine();

    }

    class Car
    {
        public string Make { get; set; }
        public string Model { get; set; }
```

```
            public int Year { get; set; }
            public string Color { get; set; }
            public double OriginalPrice { get; set; }
            /*
            public Car()
            {
                // You could load from a configuration file, a database,
etc.
                this.Make = "Nissan";
            }

            public Car(string make, string model, int year, string color)
            {
                Make = make;
                Model = model;
                Year = year;
                Color = color;
            }
            */

            public static void MyMethod()
            {
                Console.WriteLine("Called the static MyMethod");
                Console.WriteLine();
            }

        }

    }
}
```

13.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ClassesInheritance
{
    class Program
    {

        static void Main(string[] args)
        {
            car mycar = new car();
            mycar.Make = "BWM";
            mycar.Model = "745li";
            mycar.Color = "Black";
            mycar.Year = 2005;

            printVehicleDetails(mycar);

            Truck mytruck = new Truck();
            mytruck.Make = "Ford";
            mytruck.Model = "F950";
            mytruck.Year = 2006;
            mytruck.Color = "Black";
            mytruck.TowingCapacity = 1200;
            printVehicleDetails(mytruck);

            Console.ReadLine();
        }

        private static void printVehicleDetails(Vehicle vehicle)
        {
            Console.WriteLine("Here are the vehicle's details: {0}",
vehicle.FormatMe());
        }

        abstract class Vehicle
        {
            public string Make { get; set; }
            public string Model { get; set; }
            public int Year { get; set; }
            public string Color { get; set; }
```

```
            public abstract string FormatMe();
        }

        class car : Vehicle
        {

            public override string FormatMe()
            {
                return String.Format("{0} - {1} - {2} - {3}", this.Make,
this.Model, this.Color, this.Year);

            }
        }

        sealed class Truck : Vehicle
        {
            public int TowingCapacity { get; set; }

            public override string FormatMe()
            {
                return String.Format("{0} - {1} - {2} Towing units",
this.Make, this.Model, this.TowingCapacity);

            }

        }
    }
}
```

14.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace NamespacesAndReferencesAssemblies
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            //System.IO.StreamReader myStreamReader = new
System.IO.StreamReader();
            StreamReader mystreamReader = new StreamReader();
        }
    }
}
```

15.

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;

namespace AssembliesAndNamespaces
{
    class Program
    {
        static void Main(string[] args)
        {


//File.WriteAllText(@"C:\Users\Daniel\Downloads\MinigameSettings.txt",
text);


//https://msdn.microsoft.com/en-us/library/fhd1f0sw(v=vs.110).aspx
            WebClient client = new WebClient();
            string reply =
client.DownloadString("Http://msdn.microsoft.com");
```

```
            Console.WriteLine(reply);

File.WriteAllText(@"C:\Users\Daniel\Downloads\MinigameSettings.txt",
reply);

            Console.ReadLine();
        }
    }
}
```

16.

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WorkingWithCollections
{
    class Program
    {
        static void Main(string[] args)
        {
            /* commented out for object initializer below, line 77
            Car car1 = new Car();
            car1.Make = "Oldsmobile";
            car1.Modle = "Cutlas Supreme";
            car1.VIN = "A1";

            Car car2 = new Car();
            car2.Make = "Geo";
            car2.Modle = "Prism";
            car2.VIN = "B2";

            Book book1 = new Book();
            book1.Author = "Robert Tabor";
```

```
            book1.Title = "Microsoft .NET XML Web Services";
            book1.ISBN = "0=000=00000=0";
            */

            /*
            //old style collections

            //ArrayLists are dynamically sized, coolfeatures like sorting,
remove items
            ArrayList myArrayList = new ArrayList();
            myArrayList.Add(car1);
            myArrayList.Add(car2);
            myArrayList.Add(book1);
            myArrayList.Remove(book1);

            foreach (Car car in myArrayList)
            {
                Console.WriteLine(car.Make);

            }

            */


            /*
            //Newer style of collections = List<T> - generics where type
<T> customizes generic to specific

            List<Car> myList = new List<Car>();
            myList.Add(car1);
            myList.Add(car2);
            // myList.Add(book1);  --won't allow as list is specific to
car datatype

            foreach (Car car in myList)
            {
                Console.WriteLine(car.Make);
            }
            */
```

```
            /*
            //Dictionary<TKey, TValuel> is a key/value pair list
            Dictionary<string, Car> myDictionary = new Dictionary<string,
Car>();
            myDictionary.Add(car1.VIN, car1);
            myDictionary.Add(car2.VIN, car2);

            Console.WriteLine(myDictionary["B2"].Make);
            */

            //initializing an array w/o using new keyword
            //string[] names = { "Bob", "Steve", "Brian", "Chuck" };

            //object initializer  using object initialization syntax
            //No need for constructor
            //comment out car and books above to do stuff below
            Car car1 = new Car() { Make = "BMW", Modle = "750li", VIN =
"C3" };
            Car car2 = new Car() { Make = "Toyota", Modle = "4Runner", VIN
= "D4" };

            //collection initializer
            List<Car> myList = new List<Car>
            {
                new Car {Make = "Oldsmobile", Modle = "Cutless Supreme",
VIN = "A1" },
                new Car {Make = "Nisan" , Modle = "Altima", VIN = "F6" }
            };

            Console.ReadLine();

        }

        class Car
        {
            public string Make { get; set; }
            public string Modle { get; set; }
            public string VIN { get; set; }
        }
```

```
        class Book
        {
            public string Title { get; set; }
            public string Author { get; set; }
            public string ISBN { get; set; }
        }
    }
}
```

17.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace UnderstandingLINQ
{
    class Program
    {
        //2 styles : query syntax similar to sql; method synstax familiar
to c# devs
        static void Main(string[] args)
        {
            List<Car> myCars = new List<Car>() {
                new Car() { VIN="A1", Make = "BMW", Model= "550i",
StickerPrice=55000, Year=2009},
                new Car() { VIN="B2", Make="Toyota", Model="4Runner",
StickerPrice=35000, Year=2010},
                new Car() { VIN="C3", Make="BMW", Model = "745li",
StickerPrice=75000, Year=2008},
                new Car() { VIN="D4", Make="Ford", Model="Escape",
StickerPrice=25000, Year=2008},
                new Car() { VIN="E5", Make="BMW", Model="55i",
StickerPrice=57000, Year=2010}
            };
```

```
            //LINQ Query
            /*
            var bmws = from car in myCars
                       where car.Make == "BMW"
                       && car.Year == 2010
                       select car;

            */

            /*
            var orderedCars = from car in myCars
                              orderby car.Year descending
                              select car;

            */

            //LINQ method
            //lamba expressions - mini methods
            //var bmws = myCars.Where(p => p.Make == "BMW" && p.Year ==
2010);

            //var orderedCars = myCars.OrderByDescending(p => p.Year);

            /*
            var firstBWM = myCars.OrderByDescending(p => p.Year).First(p
=> p.Make == "BMW");
            Console.WriteLine(firstBWM.VIN);
            */

            // Console.WriteLine(myCars.TrueForAll(p => p.Year > 2009));

            //myCars.ForEach(p => p.StickerPrice -= 3000);
            //myCars.ForEach(p => Console.WriteLine("{0} {1:C}", p.VIN,
p.StickerPrice));

            //Console.WriteLine(myCars.Exists(p => p.Model == "745li"));

            //Console.WriteLine(myCars.Sum(p => p.StickerPrice));

            /*
```

```
            foreach (var car in orderedCars)
            {
                Console.WriteLine("{0} {1}", car.Year, car.Model,
car.VIN);
            }
            */


            /*
            Console.WriteLine(myCars.GetType());
            var orderedCars = myCars.OrderByDescending(p => p.Year);
            Console.WriteLine(orderedCars.GetType());

            var bmws = myCars.Where(p => p.Make == "BMW" && p.Year ==
2010);
            Console.WriteLine(bmws.GetType());


            var newCars = from car in myCars
                          where car.Make == "BMW"
                          && car.Year == 2010
                          select new { car.Make, car.Model }; //projection =
selection new datatypes to project at runtime
            Console.WriteLine(newCars.GetType());
            */

            Console.ReadLine();
        }
    }


    class Car
    {
        public string VIN { get; set; }
        public string Make { get; set; }
        public string Model { get; set; }
        public int Year { get; set; }
        public double StickerPrice { get; set; }
    }
}
```

18.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EnumsAndSwitch
{
    class Program
    {
        static void Main(string[] args)
        {
            List<Todo> todos = new List<Todo>()
            {
                new Todo { Description = "Task 1", EstimatedHours = 6,
Status = Status.Completed },
                new Todo { Description = "Task 2", EstimatedHours = 2,
Status = Status.InProgress },
                new Todo { Description = "Task 3", EstimatedHours = 8,
Status = Status.NotStarted },
                new Todo { Description = "Task 4", EstimatedHours = 12,
Status = Status.Deleted },
                new Todo { Description = "Task 5", EstimatedHours = 6,
Status = Status.InProgress },
                new Todo { Description = "Task 6", EstimatedHours = 2,
Status = Status.NotStarted },
                new Todo { Description = "Task 7", EstimatedHours = 14,
Status = Status.NotStarted },
                new Todo { Description = "Task 8", EstimatedHours = 8,
Status = Status.Completed },
                new Todo { Description = "Task 9", EstimatedHours = 8,
Status = Status.InProgress },
                new Todo { Description = "Task 10", EstimatedHours = 8,
Status = Status.Completed },
                new Todo { Description = "Task 11", EstimatedHours = 4,
Status = Status.NotStarted },
```

```
                new Todo { Description = "Task 12", EstimatedHours = 10,
Status = Status.Completed },
                new Todo { Description = "Task 13", EstimatedHours = 12,
Status = Status.Deleted },
                new Todo { Description = "Task 14", EstimatedHours = 6,
Status = Status.Completed }
            };


            PrintAssessment(todos);
            Console.ReadLine();

        }

        private static void PrintAssessment(List<Todo> todos)
        {
            foreach (var todo in todos)
            {
                switch (todo.Status)
                {
                    case Status.NotStarted:
                        Console.ForegroundColor = ConsoleColor.Red;
                        break;
                    case Status.InProgress:
                        Console.ForegroundColor = ConsoleColor.Green;
                        break;
                    case Status.OnHold:
                        Console.ForegroundColor = ConsoleColor.DarkRed;
                        break;
                    case Status.Completed:
                        Console.ForegroundColor = ConsoleColor.Blue;
                        break;
                    case Status.Deleted:
                        Console.ForegroundColor = ConsoleColor.Yellow;
                        break;
                    default:
                        break;
                }
                Console.WriteLine(todo.Description);
            }
```

```
        }
    }

    class Todo
    {
        public string Description { get; set; }
        public int EstimatedHours { get; set; }
        public Status Status { get; set; }
    }

    enum Status
    {
        NotStarted,
        InProgress,
        OnHold,
        Completed,
        Deleted
    }
}
```

19.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Threading.Tasks;

namespace HandlingException
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                string content = File.ReadAllText
                (@"C:\Users\Daniel\Downloads\MinigameSettings.txt");
```

```
                Console.WriteLine(content);
            }
            catch(FileNotFoundException ex)
            {
                Console.WriteLine("There was a problem!");
                Console.WriteLine("Make sure the file name is named
correctly: MinigameSettings.txt");
            }
            catch(DirectoryNotFoundException ex)
            {
                Console.WriteLine("There was a problem!");
                Console.WriteLine(@"Make sure the directory c:\Users\...
exists");
            }
            catch (Exception ex)
            {
                Console.WriteLine("There was a problem!");
                Console.WriteLine(ex.Message);
            }
            finally
            {
                Console.WriteLine("Closing the app now...");
            }
            Console.ReadLine();
        }
    }
}
```