Variables Lab

Lab 1: Declare Different Variable Types

Open the **CSharpSamples** project and open the **Program.cs** file.

char

Represents a Unicode UTF-16 character. Replace the contents of the **Program.cs** file with the following code.

```
char rating;

rating = 'A';

Console.WriteLine(rating);

Console.ReadKey();
```

Try It Out

Run the application and view the output.

string

An array of individual characters. Replace the contents of the **Program.cs** file with the following code.

```
string name;

name = "Helmet";

Console.WriteLine(name);
```

Try It Out

Run the application and view the output.

bool

Represents **true** or **false**. Replace the contents of the **Program.cs** file with the following code.

```
bool isActive;

isActive = true;

Console.WriteLine(isActive);
```

Try It Out

Run the application and view the output.

int

A 32-bit integer. Replace the contents of the **Program.cs** file with the following code.

```
int id;

id = 1;

Console.WriteLine(id);
```

Try It Out

Run the application and view the output.

decimal

A decimal is 8 bytes, ~15-17 digits. Replace the contents of the **Program.cs** file with the following code.

```
decimal price;

price = 5.99;

Console.WriteLine(price);
```

Try It Out

Run the application and view the error. Add the **'M'** suffix after the 5.99. Run the application again and view the output.

DateTime

A structure that represents a date and a time. Replace the contents of the **Program.cs** file with the following code.

```
DateTime sellDate;

sellDate = DateTime.Parse("11/1/2022");

Console.WriteLine(sellDate);
```

Try It Out

Run the application and view the output.

DateOnly

A structure that represents only a date. Replace the contents of the **Program.cs** file with the following code.

```
DateOnly endDate;

endDate = DateOnly.FromDateTime(sellDate);

Console.WriteLine(endDate);
```

Try It Out

Run the application and view the output.

Lab 2: Declare & Assign Values

Show the other numeric data types. Replace the contents of the **Program.cs** file with the following code.

```
byte data = 1;
short index = 2;
long bin = 12345678901234;
float weight = 36.24;
double cost = 1.92;

Console.WriteLine(data);
Console.WriteLine(index);
Console.WriteLine(bin);
Console.WriteLine(weight);
Console.WriteLine(cost);
```

Try It Out

Run the application and view the output.

Lab 3: The object Data Type

Replace the contents of the **Program.cs** file with the following code.

```
object value;

// Make the variable a numeric
value = 10;

Console.WriteLine(value);

// Make the variable a string
value = "Hi There";
Console.WriteLine(value);
```

Try It Out

Run the application and view the output.

Lab 4: Default Values

Replace the contents of the **Program.cs** file with the following code.

```
int id = default;
decimal price = default;
DateTime sellDate = default;
bool isActive = default;

Console.WriteLine(id);
Console.WriteLine(price);
Console.WriteLine(sellDate);
Console.WriteLine(isActive);
```

Try It Out

Run the application and view the output.

string Default

Replace the contents of the **Program.cs** file with the following code.

```
string name = default;
Console.WriteLine(name);
```

Try It Out

Notice the error on the **default** keyword. String data types are **null** by default.

Lab 5: Nullable Data Types

Replace the contents of the **Program.cs** file with the following code.

```
int? id = default;
decimal? price = default;
DateTime? sellDate = default;

Console.WriteLine(id);
Console.WriteLine(price);
Console.WriteLine(sellDate);
```

Try It Out

Run the application and view the output which should all be null values.