# Structure of C# Programs and Identifier Naming

## Identifier naming rules in C#

* + A name must consist of only letters (A-Z,a-z), digits (0-9), or underscores ( )
  + The first character must be either a letter or an underscore
  + A name can be at most 63 characters in length
  + A name must not be identical to a reserved word such as class, namespace, int, void, static

**Lesson 1.1:** Identifier Naming

Consider the following names and check if each of them is a valid name for a C# identifier.

|  |  |  |  |
| --- | --- | --- | --- |
| XXX | Y | $$$ | string |
| i j | Student ID | HelloWorld!! | first-time |

|  |  |  |
| --- | --- | --- |
| **Identifier Name** | **Valid in C#?** | **Reason** |
| *XXX* | *yes* | *not violating the naming rules* |
| *Y* | ***Yes*** | ***not violating the naming rules*** |
| *$$$* | ***No*** | ***First character cannot be $ symbol*** |
| *string* | ***No*** | ***string is reserved key*** |
| *i j* | ***No*** | ***Space is not allowed in between*** |
| *Student ID* | ***No*** | ***Space is not allowed in between*** |
| *HelloWorld!!* | ***No*** | ***Special symbols not allowed*** |
| *first-time* | ***No*** | ***Special symbols not allowed*** |

**Lesson 1.2:** Basic Structure of C# Programs

Here is an example of C# program structure.

1. namespace ( A) {
2. class ( B) {
3. static void M ain () {

4 \_\_\_( C)

5 }

6 }

7 }

From the above, the positions (A), (B), and (C) have the following meanings:

* 1. is for the namespace’s name
  2. is for the class’s name
  3. is for statements telling the computer what to do

Now complete the following tasks:

* + - Write a program that can compile and run. Your program must meet these requirements:
      * **namespace** has the name TestingNamespace
      * **class** has the name TestingClass
      * The program outputs no results

namespace TestingNamespace

{

internal class TestingClass

{

static void Main(string[] args)

{

}

}

}

* + - Write another program that can also compile and run, but now it must meet these re- quirements:
      * There is no **namespace**
      * **class** has the name SayHello
      * The program outputs a phrase *“Hello Section 451”*

internal class SayHello

{

static void Main(string[] args)

{

Console.Write("Hello Section 451");

}

}

# Data Types and Variables

## Data types in C#

|  |  |
| --- | --- |
| **Type** | **Description** |
| char | Single character |
| bool | Truth value – *true* or *false* |
| byte | Unsigned integer between 0 and 255 |
| int | Signed integer between -2,147,483,648 and 2,147,483,647 |
| uint | Unsigned integer between 0 and 4,294,967,295 |
| long | Signed integer between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807 |
| ulong | Unsigned integer between 0 and 18446744073709551615 |
| float | Real number |
| double | Double-precision real number |
| string | Sequence of characters |

**Declaration of Variables and Constants**

*Variable Declaration*

Declaring the variable myvar to be of type uint: uint var name;

Declaring the variable myvar to be of type long and have the initial value 30000: long var name = 30000;

*Constant Declaration*

Declaring the constant myconst to be of type double and have the fixed value 2.717: const double myconst = 2.717;

*Example*

const float PI = 3.414; const int FreezingPoint = 32; int X, Y;

int AREA;

char ch = ’A’;

string mynote = "Hello, Kitty"; int j = 5;

**Lesson 2.1:** Variable and Constant Declaration

Type the following code into your editor

1

using System ;

class Compute Area { static void Main () {

const double PI = 3 .1415926535 ;

radius = 12.5;

area = PI \* radius \* radius;

Console . Write Line (" Circle area = {0}", area );

}

}

2

3

4

5

6

7

8

9

10

Then try to compile the program. What errors are reported, and at which lines? What do you think is the cause?

Line 3 => no spaces allowed in class name

Line 5 => no spaces allowed between number

Line 7 => Variable radius not defined

Line 8 => Variable area not defined

Line 9 => no spaces allowed in method name

Fix the above code so that it compiles and runs. How did you fix the errors?

class ComputeArea

{

static void Main()

{

const double PI = 3.1415926535;

double radius = 12.5;

double area = PI \* radius \* radius;

Console.WriteLine(" Circle area = {0}", area);

}

}

**Lesson 2.2:** Variable Declaration (2)

Type the following code into *SharpDevelop*

using System ;

class N 22 Class

{

static void Main () { int f = 1.5 F;

char s = " Harry. Potter"; char c = ’A ’;

Console . Write Line (" f. value . ={0}, s = {1}, c = {2}", f, s, c);

}

}

Fix the above code so that it compiles and runs. How did you fix the errors?

class N22Class

{

static void Main()

{

float f = 1.5f;

string s = " Harry. Potter";

char c = 'A';

Console.WriteLine(" f. value . ={0}, s = {1}, c = {2}", f, s, c);

}

}

**Lesson 2.3:** Choosing Data Types

For each of the data item given below, choose an appropriate data type and write a C# statement to declare the variable

* Variable myAge to store your own age: Integer
* Variable income to keep track of Peter’s personal income: double
* Variable temp c to store temperature in degree Celcius: double
* Variable temp k to store temperature in Kelvin: double
* Variable name to store Aum’s full name: string

## Mathematic Expressions

Mathematic operators in C# are as follows:

|  |  |
| --- | --- |
| **Operator** | **Symbol** |
| Add | + |
| Subtract | – |
| Multiply | \* |
| Divide | / |
| Modulo (division remainder) | % |
| Grouping | ( ) |

Type the following code into *SharpDevelop*

using System ; class Test {

static void Main () { double x = 3.0 , y = 2.0;

int a = 10 , b = 2;

Console . Read Line ();

}

}

Notice that line 6 is blank. Now fill the blank with each of the statements listed in the table below, one at a time. Run the program and put the result in the right column.

|  |  |
| --- | --- |
| **Statement** | **Result** |
| Console.WriteLine(a); | 10 |
| Console.WriteLine(x+a); | **13** |
| Console.WriteLine(a/b); | **5** |
| Console.WriteLine(y/x); | **0.66666666666667** |
| Console.WriteLine(y%x); | **2** |
| Console.WriteLine((a+b)/b%a); | **6** |
| Console.WriteLine(9.0/5.0\*(a-x)); | **12.6** |
| Console.WriteLine(x+y-x\*y%x); | **5** |
| Console.WriteLine(57%50/25); | **0** |

## Output Statements

**Lesson 4.1:** Basic usage of Console.Write and Console.WriteLine

Write a C# program that outputs the following result:

Live as if you were to die tomorrow. Learn as if you were to live forever.

-Mahatma Gandhi

Then copy the code you wrote in *your editor* into the box below.

class print

{

static void Main()

{

Console.WriteLine("Live as if you were to die tomorrow.");

Console.WriteLine("Learn as if you were to live forever.");

Console.WriteLine("-Mahatma Gandhi\r\n");

}

}

**Lesson 4.2:** Advanced usage of Console.Write and Console.WriteLine

Consider the following incomplete code:

1

using System ;

class SayHi {

static void Main () {

string yourName = (1) ; uint yourAge = (2) ;

Console . Write Line (" Hello {1}... You . are{0} years. old .", (3) , (4) );

}

}

2

3

4

5

6

7

8

9

10

Fill in the blanks (1),..., (4) so that the program will say hello to you and print out your own age. For example, if your name is *“Arthur”* and you are 18 years old, the result should be:

Hello Arthur. You are 18 years old.

What did you fill in for each of the blanks?

|  |  |
| --- | --- |
| **Position** | **Your answer** |
| (1) | **“Arthur”** |
| (2) | **18** |
| (3) | **yourName** |
| (4) | **yourAge** |