Variable types

Page 35

A *variable* **stores** a value during the running of a program

Type	Description
string	contains alpha numeric characters, persons name
int	a number value without decimal points, persons age
double	a number with decimal points, an amount of money
boolean	either true or false

c# is a *strongly typed* language. All variables **must** have a type.
We cannot "mix" types.

Simple Print to Console

Page 45

Examples

```
Console.WriteLine("Hello World");
Console.Write("and Hello Moon");
```

Line	Description
1	insert a carriage return and leave cursor on next line
2	leave cursor on same line

Placeholders

Page 49

$"{varName1} {varName2}"

Examples

```
Console.WriteLine(${"Name : {studentName}");
Console.WriteLine($"GPA : {studentGpa}");
```

Line	Description
1	**{** and **}** enclose the variable name to display

You can use multiple placeholders in one string.

Logical operators

Page 62

Operand	Description
A==B	checks two operands to see if equal. false
A!=B	check operands to see if not equal. true

Logical operators (cont)

A<B	is A less than B? false
A>=B	is A greater than or equal to B. true
A<=B	is A less than or equal to B? false

An operand is variable or value involved in operation. In examples above - A=10, B=5

Loops - While Loop

Pages 79-83,86-87

```
/*
* loop through a statement block 10 times
* if condition is not satisfied, statements will not be executed
*/
int counter=0;
while (counter <=10)
{
Console.WriteLine($"Counter value is {counter}");
counter ++;
}
```

Methods - declaration

Page 93

[static] [public|private] *return-type MethodName* ([*param-list*])

Type	Description
[static]	no need to create instance, call directly
[*public* | *private*]	can only be called from within this class
[*void* | *int* | *double* | *string*]	return type, void infers nothing returned
MethodName	use PascalCase for naming the method
([*param-list*])	specify paramater type, separate with commas

Tips n' Tricks for CA #1

Variable Definition & Assignment

Pages 43-45

[type] <varName> = <value>;

Examples

```
int studentAge=19;
string studentName="Walter";
double studentGpa=78.68;
boolean studentRegistered;
```

Item	Description
type	common types **int**, **string**, **double**, **boolean**
<varName>	the *name* of variable in which we store value

studentRegisterd is not *initialised* at declaration time above. it could be true or false.

Read Keyboard Input - Strings

Page 54

Example

```
studentName=Console.ReadLine();
```

Line	Description
1	read input from keyboard, assign to studentName

Console.ReadLine() is a method without parameters.
It takes input from the keyboard as a string

Common Formatting Codes

Page 58-59

$"{x:c} {y:p} {z:n3}"

Code	Format	Output
C or c	currency	€1,245.44
P or p	percent	4.00%
N or n	Number	103,423.346

Formatting improve the output for the user.

Above *x*=1245.443, *y*=0.04 and *z*=10342-3.3456.

If stements - combining expressions

Page 64-65

```
if ((condition1 && condtion2)
{
// execute if condition1 AND condition2 true
}
else if ((condition3 II condtion4)
{
// execute if condition3 OR condition4
```

Tip	Reason
Comment your code	allows you or someone else to more *easily* understand the code now or in the future
Watch your variables and constant naming convention	Use camelCase for ordinary variables, and UPPERCASE for constants - it's *easy* then to tell them apart

```
}
else
// then execute this statement
```

By **programming1**

cheatography.com/programming1/

Published 24th November, 2022.
Last updated 30th November, 2022.

Loops - Do While

Page 84

```
/*
* loop through a statement block 10 times
* statement block will always execute at least once
* even if counter was initially 11!!
*/
int counter=0;
do
{
Console.WriteLine($"Counter value is {counter}");
counter ++; //
} while (counter <=10)
```

Methods - related terminology

Page 92

Item	Description
return type	a method can return a value - of type int, double, string. if nothing returned, then *return-type* is void
sharing data between methods	*parameters* - values passed to a method call. also known as arguments. *class level variables* - available to all methods, scope is *global*
calling a method	We must call the method to invoke it.
predefined methods	Includes Console.WriteLine("-Hello) - has parameters Console.Readline() - no parameters

Tips n' Tricks for CA #2

Tip	Reason
Use code indent ation	When using conditional (If), Loops (While, Do While, For) and Methods - indent your code. Make it *easier* to read for everyone.
Follow the recipe	Make your input and output actually *look like* what is presented on the CA question.

Using naming conventions, comments

Pages 39,53,93,30

Type	Description
variables	camelCase, first letter is lowercase, other words first letter uppercase
constants	use uppercase, e.g. VATRATE
methods	PascalCase, first letter of each word is uppercase
comment our code	// what does this code do? /* reminds colleagues, our future selves ☺ */

Coding conventions are important within a team.

It is part of the common language of writing software code.

Read Keyboard input - Numbers

Page 55

Example

```
studentAge = int.Parse(Console.ReadLine());
studentGrade = double.Parse(Console.ReadLine());
```

Line	Description
1	Integer value is returned and assigned
2	Double value may contain decimals

Extract a number from the keyboard input with .Parse()

Neater Printing in Tables

Page 57

$"<text>{<expression>,<field-width>}<text>..."

Examples

```
$"Name :{studentName,20}"
$"{"Name",-20} :{studentName}"
```

Line	Description
1	right justify, 20 leading spaces before student name
2	left justify, 20 spaces after label "-Name"

If statements - Examples

Pages 66-73

```
if (studentGpa>=70)
{
Console.WriteLine("Honours");
}
else if (studentGpa<70 && studentGpa>=50)
{
Console.WriteLine("Distinction");
}
else Console.WriteLine("Fail");
```

Loops - For Loop

Pages 85,87

```
/*
* initial value of counter set in for statement
* counter is incremented then statement block complete
*/
int counter;
for (counter=0; counter <=10; counter++)
{
Console.WriteLine($"Counter value is {counter}");
}
```

Methods - full example

```
class Program
{
  static void Main(s tring[]
args)
 {
    static string saluta tio -
n="H ell o";
    string name=G etN ame();
    Pri ntG ree tin g(n ame);
 }
  static private string
GetName()
 {
      Con sol e.W rit eLi -
ne( " Enter First Name : ");
      string firstN ame =na -
meC ons ole.Re adL ine();
      return firstName;
 }
static private void PrintG ree -
tin g(s tring name)
 {
    Con sol e.W rit eLi ne( $"
{s alu tation} {name} !");
 }
}} // end class
```

Bits and pieces

Console.OutputEncoding=System.Text.E-ncoding.UTF8; // display special symbols like currency

Carriage Return or "\n" // A *carriage return* moves the cursor onto the next line in our console display.

Lab worksheet is a solution. Each question is a project