Antony Roshan

**C SHARP**

* Developed by Anders Hejlsberg and team
* Language Designed for CLI

**Features of C#**

* Strongly typed
* Object oriented
* Component oriented
* Multi-paradigm
* Event driven
* Task driven

**Datatype**

* **Value types**

Stores value on the stack

Classified into two:

* Structs
* Enums

The following data types are all of value type:

* bool
* byte
* char
* decimal
* double
* enum
* float
* int
* long
* sbyte
* short
* struct
* uint
* ulong
* ushort
* **Reference** **types**

Stores reference on the stack and later store the value on the heap

The followings are reference type data types:

* String
* Arrays
* Object
* Class
* Interface
* Delegate

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

int age = 10;

string name = "Roshan";

Console.WriteLine("Name is " + name + "\nAge is " + age);

Console.WriteLine("Username is {0} \nUser Age is {1}",name,age);

}

}

}

**Output**

Name is Roshan

Age is 10

Username is Roshan

User Age is 10

Press any key to continue . . .

**Type casting**

* Convert one datatype into another

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

short num1 = 10;

short num2 = 20;

short sum = (short)(num1 + num2);

Console.WriteLine(sum);

}

}

}

**Output**

30

Press any key to continue . . .

**Boxing & Unboxing**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

int a = 10;

object o = a; //boxing

int b = (int)o; //unboxing

}

}

}

**Nullable Type**

* To store null value in value type
* If data is coming from a database there may be null value
* So we can use nullable type

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

string s = null; //Reference type

int? a = 10; //a = 10 //value type

int? b = a ?? 20;

Console.WriteLine(b);

int? c = null; //c = null //value type

int? d = c ?? 20;

Console.WriteLine(d);

}

}

}

**Output**

10

20

Press any key to continue . . .

**Operators**

* **Arithmetic Operator**

Commonly used operators are:

+

-

\*

/

%

++ and – are short hand operators

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

int a = 50, b = 10,c = 5;

Console.WriteLine(a + b);

Console.WriteLine(a - b);

Console.WriteLine(a \* b);

Console.WriteLine(a / b);

Console.WriteLine(a % b);

int d = a + b \* c;

Console.WriteLine(d);

int x = a++;

Console.WriteLine(a);

Console.WriteLine(x);

}

}

}

**Output**

60

40

500

5

0

100

51

50

* **Relational/Comparison Operator**

Commonly used operators are:

==

!=

>

<

>=

<=

Return a Boolean value after comparing

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

int a = 50, b = 10;

Console.WriteLine(a > b); //true

Console.WriteLine(a < b); //false

Console.WriteLine(a >= b); //true

Console.WriteLine(a <= b); //false

Console.WriteLine(a == b); //false

Console.WriteLine(a != b); //true

int num = 46;

string s = num % 2 == 0 ? "Even" : "Odd";

Console.WriteLine(s);

}

}

}

**Output**

True

False

True

False

False

True

Even

Press any key to continue . . .

* **Logical Operator**

Commonly used operators are:

&&

||

!

Return a Boolean value after comparing

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("AND");

Console.WriteLine("--------------------------");

Console.WriteLine(true && true);

Console.WriteLine(true && false);

Console.WriteLine(false && true);

Console.WriteLine(false && false);

Console.WriteLine("OR");

Console.WriteLine("==========================");

Console.WriteLine("--------------------------");

Console.WriteLine(true || true);

Console.WriteLine(true || false);

Console.WriteLine(false || true);

Console.WriteLine(false || false);

}

}

}

**Output**

AND

--------------------------

True

False

False

False

OR

==========================

--------------------------

True

True

True

False

Press any key to continue . . .

**Control Statements**

* Control flow of execution
* Enable statement to execute on specific condition
* Enable statements to be executed repeatedly
* Enable to jump on particular point of code
* Can be classified into:
* Conditional statements
* Iteration statements /loops
* Jump statements

**Conditional Statements**

Only execute if the condition is true

* **If else statement**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a number");

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

if(num % 2 == 0)

{

Console.WriteLine("Number is Even");

}

else

{

Console.WriteLine("Number is Odd");

}

}

}

}

**Output**

Enter a number

26

Number is Even

Press any key to continue . . .

* **Switch statement**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a character:");

char ch = char.Parse(Console.ReadLine());

string s = ch.ToString().ToLower();

switch(s)

{

case "a":

Console.WriteLine("a is vowel");

break;

case "e":

Console.WriteLine("e is vowel");

break;

case "i":

Console.WriteLine("i is vowel");

break;

case "o":

Console.WriteLine("o is vowel");

break;

case "u":

Console.WriteLine("u is vowel");

break;

default:

Console.WriteLine("Constant / Special characters");

break;

}

}

}

}

**Output**

Enter a character:

o

o is vowel

Press any key to continue . . .

**Iteration statement / Loops**

Can be classified into:

* For loop
* While loop
* Do while loop
* Foreach loop
* **For loop**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

int sum = 0;

for(int i=1; i <= 10; i++)

{

Console.WriteLine(i);

sum = sum + i;

}

Console.WriteLine("Sum is {0}",sum);

}

}

}

**Output**

1

2

3

4

5

6

7

8

9

10

Sum is 55

Press any key to continue . . .

* **While loop**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

bool isprime = true;

int count = 0;

int num = 1;

while (count < 100)

{

isprime = true;

for(int i = 2; i <= Math.Sqrt(num); i++)

{

if(num % i == 0)

{

isprime =false;

break;

}

}

if(isprime)

{

Console.Write(num + "\t");

count++;

}

num++;

}

Console.WriteLine();

}

}

}

**Output**

1 2 3 5 7 11 13 17 19 23 29 31 37 41 43

47 53 59 61 67 71 73 79 83 89 97 101 103 107

109 113 127 131 137 139 149 151 157 163 167 173 179 181

191 193 197 199 211 223 227 229 233 239 241 251 257 263

269 271 277 281 283 293 307 311 313 317 331 337 347 349

353 359 367 373 379 383 389 397 401 409 419 421 431 433

439 443 449 457 461 463 467 479 487 491 499 503 509 521

523

Press any key to continue . . .

* **Do while loop**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

string actualpin = "4321";

int count = 0;

string pin;

do

{

Console.WriteLine("Enter a pin:");

pin = Console.ReadLine();

count++;

}

while (pin != actualpin && count < 3);

if(pin == actualpin)

{

Console.WriteLine("Welcome user");

}

else

{

Console.WriteLine("Account locked");

}

}

}

}

**Output**

Enter a pin:

1568

Enter a pin:

3984

Enter a pin:

4563

Account locked

Press any key to continue . . .

**Jump statement**

Can be classified into:

* Break
* Continue
* Goto
* Return
* **Break statement**

Used to terminate loops and switch statement

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

for(int i=1;i<10;i++)

{

if(i==5)

{

break;

}

Console.WriteLine("i = "+i);

}

}

}

}

**Output**

i = 1

i = 2

i = 3

i = 4

Press any key to continue . . .

* **Continue statement**

It forces the next iteration of the loop to take place , skipping any code in between

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

for(int i=1;i<10;i++)

{

if(i==5)

{

continue;

}

Console.WriteLine("i = "+i);

}

}

}

}

**Output**

i = 1

i = 2

i = 3

i = 4

i = 6

i = 7

i = 8

i = 9

Press any key to continue . . .

* **Goto statement**

Provides an unconditional jump from the ‘goto’ to a labelled statement in the same function

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

int i = 1;

abc:

Console.WriteLine(i);

i++;

if (i <= 10)

goto abc;

}

}

}

**Output**

1

2

3

4

5

6

7

8

9

10

Press any key to continue . . .

* **Return statement**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter first number:");

int num1 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter second number:");

int num2 = int.Parse(Console.ReadLine());

if(num2 == 0)

{

Console.WriteLine("Cannot divide the number by 0");

return;

}

int result = num1 / num2;

Console.WriteLine("Result is :" + result);

}

}

}

**Output**

Enter first number:

25

Enter second number:

0

Cannot divide the number by 0

Press any key to continue . . .