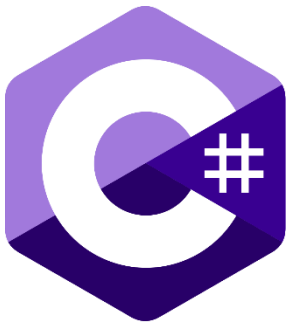
Code Clock

DAY 3: Methods

Learn.to.code



Programming with C#

@ QUB

Methods

A method (called a ***function*** in Python) is a reusable block of code that performs a specific task. Functions help us to write clean, organised, and reusable code.

C# has many built-in methods like `Console.WriteLine()`, but we can also write our own. These are called **user-defined methods**.

Defining a Method

To define a method in C#, you declare its return type, name, and any parameters it uses. Methods must be placed inside a class.

Example:

```
void CodeClock()
{
    Console.WriteLine("Welcome to Code Clock Day 3");
}
```

In the **main method *CodeClock***(); // Calls the method

```
public static void Main (string [] args)
{
  CodeClock();
}
```

Method Parameters

Methods can accept one or more ***parameters*** (also called arguments) to make them more flexible.

Example:

```
void PrintMyName(string name)
{
    Console.WriteLine("Welcome to Code Clock Week 4 " + name);
}
```

In the **main method *PrintMyName***(“Errol”); // Calls the method

```
public static void Main (string [] args)
{
  PrintMyName("Errol");
}
```

C# is **strongly typed**, so you must always declare the data type of the parameter.

Multiple Parameters:

```
void PrintNames(string name1, string name2, string name3)
{
    Console.WriteLine("Welcome to Code Clock Week 4\n" + name1 + "\n" + name2 + "\n" + name3);
}
```

In the **main method *PrintNames***(“David”, “Michael”, “Bradley”); // Calls the method

```
public static void Main (string [] args)
{
  PrintNames("David", "Michael", "Bradley");
}
```

Different Data Types:

```
void PrintTrainerDetails(string name, string age, string gender)
{
    Console.WriteLine("Your trainer today is:\n- Name: " + name + "\n- Age: " + age + "\n- Gender: " +
gender);
}
```

In the **main method *PrintTrainerDetails***(“Errol Martin”, “45”, “Male”); // Calls the method

```
public static void Main (string [] args)
{
    PrintTrainerDetails("Errol Martin", "45", "Male");
}
```

```
```

Unknown Number of Parameters

In C#, this can be done using the `params` keyword:

```
void PrintNames(params string[] names)
{
    foreach (string name in names)
    {
        Console.WriteLine(name);
    }
}
```

In the **main method *PrintNames***(“Rosie”, “Sam”, “Autumn”); // Calls the method

```
public static void Main (string [] args)
{
    PrintNames("Rosie", "Sam", "Autumn");
}
```

Default Parameters

C# allows default values for parameters:

```
void PrintTrainerDetails(string name, string age, string gender, string course = "Programming with C#")
{
    Console.WriteLine("Name: " + name);
    Console.WriteLine("Age: " + age);
    Console.WriteLine("Gender: " + gender);
    Console.WriteLine("Course: " + course);
}
```

In the **main method *PrintTrainerDetails***(“Errol Martin”, “45”, “Male”); // Calls the method

```
public static void Main (string [] args)
{
  PrintTrainerDetails("Errol Martin", "45", "Male");
  PrintTrainerDetails("Errol Martin", "45", "Male", "Programming with Python");
}
```

Return Statement (single value)

```
int SimpleCalc(int num1, int num2)
{
    return num1 + num2;
}
```

In the **main method** ***SimpleCal can be appended to the end of Console.WriteLine statement***

```
public static void Main (string [] args)
{
  Console.WriteLine("The sum of 12 and 14 is " + SimpleCalc(12, 14));
}
```

```

# Challenge:

1. Write a method and call it, which prints a parameter containing today's date i.e. 25th Feb 2023
2. Write a method, and call it, which when called prints the number 1 to 10
3. Write a method, and call it, which calculates and prints the average of two numbers passed as parameters.
4. 
   a) Write a method, and call it, which takes three integer parameters and calculates the volume of a rectangle, returning both the volume and unit of measurement i.e. "$cm^3$" and prints in a suitable output statement.
   b) Adapt the above to enable the user to enter the three dimensions.
5. 
   a) Write a method which calculates and returns both the simple interest AND amount to repay to replay on a user-defined loan amount and user-defined number of years with an interest rate of 4.5%. The code should ensure that the user enters a loan amount of no more than £30,000 and no less than £1000. The minimum period should also be no less than 2 years. Both values should be printed within the function call statement.
   b) Create another method which now calculates compound interest for both user-defined loan amount and number of years.
   c) Adapt the code to enable the user to choose whether to calculate either simple interest or compound interest.
   d) Adapt the code to enable the user to calculate either simple or compound interest more than once.
   e) Create a menu which enables the user to choose either:
      1. Calculate Simple Interest
      2. Calculate Compound Interest
      3. Stats*
      This menu will enable the user to choose which function to call.
      *This should print out the number of times the interest functions have been called during that a particular session.
   f) Adapt the code to create another method which handles after the input of these values
```

6. Create a registration method which enables an user to register within an application using their
   - first name,
   - surname,
   - date of birth and
   - gender.

   This registration function must also:

   - validate the password to ensure it meets the following criteria.
     - must not contain the word password
     - must be between 9 and 15 characters in length
     - must contain a Upper case letter
     - must contain a Lower case letter
     - must contain one of the following special characters:
     - @
     - #
     - £
     - &
     - ?

     The registration method should also call a method which generates an username which uses the first initial of the registrants first name, their surname and randomly generate 3-digit number to create a username in the form “asmith123”. The user should receive confirmation of both their newly generated username and also confirmation that the password was successfully.