METHODS

WHAT IS A METHOD?

A method is a sequence of instructions that performs a specific task, and those instructions are packaged as a single unit.

DESCRIPTION

- To allow other classes to access a method, use the public keyword. To prevent other classes from accessing it, use the private keyword.
- Methods that don’t return any data should have the void keyword as the return type.

DESCRIPTION

- Names of methods should typically start with verbs. Two very commonly used examples of this are methods which either set or return the values of instance variables, called setters and getters respectively.

SYNTAX

```
public returnType methodName([parameterList])
{
    statements of the method
}
```

EXAMPLE

```
public String getCode()
{
    String code="Java 101";
    return code;
}
```

METHOD OVERLOADING

When you create two or more methods with the same name but with different:

- Parameter type
- Parameter order
- Parameter count
- Passing method (in, out, ref)

CALLING METHODS

- When calling a method with no arguments, we include an empty set of parentheses after the method name.
- If a method returns a value, we can code an assignment statement to assign the return value to a variable. The data type of that variable must be compatible with the data type of the return value.

SYNTAX

```
objectName.methodName(argumentList)
```

EXAMPLES

```
product.printToConsole();

product.setCode(productCode);

double price = product.getPrice();
```

PASSING PARAMETERS TO METHODS

There are three ways to send parameters to methods in C#

- Pass by value
- Pass by reference
- Pass by out

RECURSION

- Recursion is a concept that aims to solve a problem by dividing it into smaller chunks in what is called divide and conquer.
- The idea is that a method can call itself, but with parameters that represent a smaller instance of the problem.
- You need to specify a stop condition to end the recursion.

EXAMPLE!

- Do the Factorial example.

RECAP

WHAT YOU SHOULD KNOW AT THIS POINT:

- What are methods and why are they important
- Syntax for writing methods
- How to do method overloading
- How to call methods
- Passing parameters to methods
- Doing recursion