Connor Petford (Student)

Connor.petford@techtorium.ac.nz

SR1.1 Document

Console Application Development

procedural and object- oriented programming/

Number and Coding system

# Procedural Programming

## Concepts

Procedural programming is defined as a group of programming languages, based on the idea of calling procedures, derived from structured programming. Procedures, usually referred to as routines or functions, are essentially a set of computing processes that must be executed. Every given process may be called at any time while a programme is running, either by other procedures or the procedure itself. A few *examples* of procedural programming languages are C, C++ (also part of object-oriented), Pascal, Basic, and FORTRAN.

## Key Ideas

A few key ideas relating to procedural programming:

Procedural programming introduced what we call module functions, or operations that can be called to execute at any time by any running procedure or the procedure itself.

Functions can be coded and reused anywhere in your project, by calling the function name, and the code under the function will be executed

Procedural programming however does not support encapsulation. It also does not support events.

# Object-Oriented Programming

## Concepts

Object-oriented programming is defined as a group of programming languages, based on the idea of objects. In the form of properties and methods, objects contain both data and code. Computer programmes are created utilising the idea of objects that interact with the outside environment in object-oriented programming. Although there are many different object-oriented programming languages, the most widely used ones are class-based, which means that objects are instances of classes that also define their types. A few *examples* of object-oriented programming languages are C#, C++ (also part of procedural programming), JavaScript, and Python.

## Key Ideas

A few key ideas relating to object-oriented programming:

Object oriented programming introduced the concepts of classes and objects. We can use these to group objects into classes which may interact with our program in different ways, therefore the different classes.

It provides a more efficient way to reuse the code using Inheritance. This is done by implementing the objects to be called/inherited when needed to be reused. This allows for us to efficiently reuse code more practically and increase productivity in our projects.

It supports encapsulation.

It allows us to create events such as key-down and button-click. These are so that we are able to call events/objects to interact how they’re coded to, on the press of a key or click of a button. This allows the user to call the object when they wish, rather than the object only being called at a certain time when the procedure calls it.

# Main Differences

Procedural programming runs on the concepts of calling procedures or functions, where a set of computer processes will be executed.

Object oriented programming runs on the concepts of objects such as properties and methods. These are class-based, and contain computer processes, but also data that can be called.

This means that procedural programming calls its processes as the procedure is running, whereas object-oriented programming is able to group different processes into classes and call them through events as well, such as key-down or button-click.

Object oriented programming supports encapsulation, whereas procedural programming doesn’t.

Object oriented programming also supports events such as the key-down and button-click, so the user is able to call an object at any given time, as opposed to procedural where it essentially follows its procedure and is called when the procedure runs.

# Number Systems

## Concepts

The programming system is broken down into numbers and coding systems, and this is how it communicates with the machine. We, as developers, understand the coding system, which we utilise and implement in our projects in Microsoft Visual Studio. This however, must be converted into number systems, as computer systems are only able to understand numbers systems. To put it simply, our processors convert our programming languages (coding systems) into binary (number systems), so our devices can output the desired code.

## Numbers Systems

The numbers systems are simply put to represent numbers. We mostly understand and use the base-10 decimal system, however there are other systems such as the base-16 hexadecimal system. There’s also the base-8 octal system, and base-2, which we know as binary. The most commonly used number systems for programming are binary, as it is easy to process.

## Coding Systems

Creating apps, websites and software requires knowledge of coding languages that basically tell the computer what it should do, and they process that information in binary numbers as they only understand number systems. ASCII and Unicode are two ways to represent text in computers.

## The Need for Each System (With Scenario)

As we are only able to efficiently understand the coding system due to it being in English, we require the use of programs that support the coding system, so we are able to understand that we are implementing the code that is required of us. However, the computer is unable to understand coding languages, so is required to process it into number systems. When the coding system is processed into number systems, the computer is able to understand number systems such as binary and is able to convert and output our program. For instance, if we wish to implement an application, we will utilise programs such as Microsoft Visual Studio, and begin to implement our application. This is done with the use of the coding system. Depending on what we may be implementing, we may use different coding languages such as C# or JavaScript. Here we understand and implement the code, and the program processes this into number systems such as binary, so the computer is able to output the application. Without the coding systems in place we would be unable to understand how to implement applications. Without the number systems our programs wouldn’t be able to process and convert our coding systems into number systems, so our computer wouldn’t understand and wouldn’t output the application.