

Procedural, Object-oriented & Event-driven programming

What are procedural, object orientated and event driven programming?

Procedural programming is the simplest type of programming – each instruction is executed one after the other (although there can be loops and if statements)

Object orientated programming (OOP) uses code ‘objects’ which have properties and can give or return variables. Each part of the program (such as a button on a form) is an object, so they can all react independently.

Event driven programming depends on the user – code is only executed when the user provides input, such as clicking or moving the mouse, or filling in a form. JavaScript often uses this paradigm.

What are these paradigms used for?

Procedural programming is often used in very simple programs, such as a program to do some basic maths, or the code that drives a robot in an assembly factory.

Object-oriented code is used for complex programs, such as games and all but the simplest of software, in which it is easier to make each ‘thing’ an object – such as a character or buttons on a form. This allows for intricate and reactive code that can do everything the other paradigms can do, and much more.

Event driven programming is used for static applications that rely heavily on user input, such as web forms.

What are their limitations?

Procedural programming is very powerful for simple or recursive tasks. It is not good for handling lots of things at once though.

OOP allows for very complex programs, such as modern video games and human-machine interfaces. However, it is difficult to learn and understand, and not all languages support OOP.

Event-driven programming is ideal for simple games and forms. It does not need much processing power. However, it relies heavily on user input, so is not good for autonomous tasks.