Titre

Etudiant 1

Etudaint 2

Titre 2

Etudiant 3

**History of object orientation – programming languages**

The first object-oriented programming language was *Simula-67*. This language allows programmers to simulate the way objects behave in the real world. For example, a simulation application might model cars approaching an intersection controlled by traffic lights. The objects in this simulation would include cars, lights and traffic lanes. When running a Simula program, each object is represented by a ‘chunk’ of data. All the procedures that operate on that object are found together in a class, so that the programmer can easily change the behavior of a car or a traffic light without having to search through the entire program.

Although Simula-67 was intended as a special-purpose simulation language, software developers gradually recognized that a wide variety of programs would be easier to develop and understand if organized this way. Although Simula is still used today, mostly in Scandinavia, it never gained widespread popularity.

In the early 1980s a new object-oriented language called *Smalltalk* gained popularity. Smalltalk was developed at Xerox PARC (Palo Alto Research Center). This research lab is also credited with giving rise to many other inventions, which we take for granted today: graphical user interfaces, the mouse, the laser printer, etc.

Smalltalk has many features that were innovative at the time. It has a simple syntax that is quite unlike that of other popular languages. It has a large library of reusable code – and programmers have access to all the source code for the library. Smalltalk popularized bytecode, platform independence and garbage collection, as now found in Java.

Smalltalk is still used today and has a loyal following, but it was rapidly

overtaken in the late 1980s by a new language called *C++.* The developer of C++, Bjarne Stroustrup, recognized the advantage of object orientation but also recognized that there were tremendous numbers of programmers of the C language who wanted to take advantage of their C expertise and C’s execution speed. He thus added object-oriented extensions to C and the new language became rapidly dominant.

However, over 15 years of experience has shown that C++ has certain drawbacks. Its syntax is quite complex and it is too easy to create code that has bugs. Large C++ programs have thus been found to be hard to maintain – they deteriorate rapidly as many programmers make changes.

In 1991, a group of engineers at Sun Microsystems started a project to design a programming language that could be used in consumer ‘smart devices’. Knowing the strengths and weaknesses of C++, Smalltalk, and a third language called *Objective-C*, they invented a language initially called Oak. This borrowed the C syntax from C++, and many of its other essential features from Smalltalk. Some of the more troublesome features of C++, such as multiple inheritance and the ability to create pointers to arbitrary parts of memory, were eliminated.

Unfortunately, the team faced difficulties trying to sell Oak. It was only when the Internet gained popularity, with the advent of the World Wide Web in 1994, that Sun saw an opportunity to exploit the technology. The new language, renamed Java, was formally presented in 1995 at the SunWorld ’95 conference.

More recently, Microsoft has entered the fray with its language *C#* (C-Sharp). C# has very many similarities with Java, but some subtle and interesting differences. C# is one of several languages that can run on Microsoft’s *Common Language Runtime*, and is part of its *.Net* framework. Anyone who knows Java should be able to learn C# quite easily.

We will continue this history of object orientation in Chapter 5, where we will look at methods and notations for describing object-oriented systems.