

# *Object-Oriented Paradigm*

1.	Objectives .....	2
2.	A little Bit of History.....	3
3.	Overview .....	4
4.	Objects .....	5
5.	Messages.....	6
6.	Classes .....	7
7.	Classes and Objects.....	8
8.	Object-oriented characteristics.....	9
9.	Implementation of Object-Oriented Constructs.....	15
10.	Dynamic Binding .....	18
11.	OO Programming Languages .....	20
11.1.	C++ .....	20
11.2.	Java .....	24
11.3.	Features Removed from C and C++ .....	<b>Error!</b>
	<b>Bookmark not defined.</b>	
11.4.	Python.....	30

## 1. Objectives

- Object technology helps build complex applications quickly: applications are created from existing components
- Easier design: designer looks at objects as a black box, not concerned with the detail inside
- Software Reuse: Classes are designed so they can be reused in many systems or create modified classes using inheritance
- Enhance the software product quality: Reliability, Productivity, etc
- Support for software adaptability more easily
- Support the move towards distributed systems
- Robustness - classes designed for repeated reuse become stable over time

## 2. A little Bit of History

- Object-oriented paradigm has its roots back to **Simula** (Simulation of real systems) language that was developed by in 1960 by researchers at the Norwegian Computing Center.
- In 1970, Alan Kay and his research group at Xerox PARK (the Palo Alto Research Center) developed the first pure OOPL: **Smalltalk**.
- In parallel to Alan's work, Bjarne Stroustrup at Bell Laboratories was also developing an extension to C language, called **C++**, that implements the OO concepts.
- By the time of the first major conference on OO programming, in 1986, there were dozens of languages: Object C (1986), Eiffel (1988), Object Pascal, etc.