

Contents

	Foreword	11
1	An Introduction to Object Orientated Programming	12
1.1	A Brief History of Computing	13
1.2	Different Programming Paradigms	14
1.3	Why use the Object Orientation Paradigm?	16
1.4	Object Oriented Principles	16
1.5	What Exactly is Object Oriented Programming?	20
1.6	The Benefits of the Object Oriented Programming Approach	23
1.7	Software Implementation	23
1.8	An Introduction to the .NET Framework	27
1.9	Summary	30
2	The Unified Modelling Language (UML)	31
2.1	An Introduction to UML	31
2.2	UML Class diagrams	32
2.3	UML Syntax	36
2.4	UML Package Diagrams	50
2.5	UML Object Diagrams	56

www.sylvania.com

**We do not reinvent
the wheel we reinvent
light.**

Fascinating lighting offers an infinite spectrum of possibilities: Innovative technologies and new markets provide both opportunities and challenges. An environment in which your expertise is in high demand. Enjoy the supportive working atmosphere within our global group and benefit from international career paths. Implement sustainable ideas in close cooperation with other specialists and contribute to influencing our future. Come and join us in reinventing light every day.

Light is OSRAM

**OSRAM
SYLVANIA**

2.6	UML Sequence Diagrams	58
2.7	Summary	59
3	Inheritance and Method Overriding	60
3.1	Object Families	61
3.2	Generalisation and Specialisation	61
3.3	Inheritance	63
3.4	Implementing Inheritance in C#	70
3.5	Constructors	70
3.6	Constructor Rules	72
3.7	Access Control	73
3.8	Abstract Classes	77
3.9	Overriding Methods	78
3.10	The 'Object' Class	80
3.11	Overriding ToString() defined in 'Object'	81
3.12	Summary	82
4	Object Roles and the Importance of Polymorphism	84
4.1	Class Types	84
4.2	Substitutability	87
4.3	Polymorphism	88
4.4	Extensibility	89




360°
thinking.

Deloitte.


Discover the truth at www.deloitte.ca/careers

© Deloitte & Touche LLP and affiliated entities.

4.5	Interfaces	96
4.6	Extensibility Again	102
4.7	Distinguishing Subclasses	105
4.8	Summary	107
5	Overloading	108
5.1	Overloading	108
5.2	Overloading To Aid Flexibility	109
5.3	Summary	112
6	Object Oriented Software Analysis and Design	113
6.1	Requirements Analysis	113
6.2	The Problem	115
6.3	Listing Nouns and Verbs	116
6.4	Identifying Things Outside The Scope of The System	117
6.5	Identifying Synonyms	118
6.6	Identifying Potential Classes	119
6.7	Identifying Potential Attributes	121
6.8	Identifying Potential Methods	121
6.9	Identifying Common Characteristics	122
6.10	Refining Our Design using CRC Cards	123

SIMPLY CLEVER **ŠKODA**


We will turn your CV into an opportunity of a lifetime



Do you like cars? Would you like to be a part of a successful brand? We will appreciate and reward both your enthusiasm and talent. Send us your CV. You will be surprised where it can take you.

Send us your CV on www.employerforlife.com



6.11	Elaborating Classes	125
6.12	Summary	126
7	Generic Collections and how to Serialize them	128
7.1	An Introduction to Generic Methods	128
7.2	An Introduction to Collections	133
7.3	Different Types of Collections	134
7.4	Lists	134
7.5	HashSets	135
7.6	Dictionaries	136
7.7	A Simple List Example	138
7.8	A More Realistic Example Using Lists	140
7.9	An Example Using Sets	145
7.10	An Example Using Dictionaries	154
7.11	Serializing and De-serializing Collections	160
7.12	Summary	165
8	C# Development Tools	166
8.1	Tools for Writing C# Programs	166
8.2	Microsoft Visual Studio	167
8.3	SharpDevelop	168
8.4	Automatic Documentation	169

I joined MITAS because
I wanted **real responsibility**

The Graduate Programme
for Engineers and Geoscientists
www.discovermitas.com



Month 16

I was a construction
supervisor in
the North Sea
advising and
helping foremen
solve problems

Real work
International opportunities
Three work placements



 **MAERSK**



8.5	Sandcastle Help File Builder	172
8.6	GhostDoc	173
8.6	Adding Namespace Comments	173
8.8	Summary	175
9	Creating And Using Exceptions	177
9.1	Understanding the Importance of Exceptions	177
9.2	Kinds of Exception	180
9.3	Extending the ApplicationException Class	180
9.4	Throwing Exceptions	182
9.5	Catching Exceptions	184
9.6	Summary	184
10	Agile Programming	185
10.1	Agile Approaches	186
10.2	Refactoring	186
10.4	Support for Refactoring	187
10.5	Unit Testing	188
10.6	Automated Unit Testing	188
10.7	Regression Testing	190
10.8	Unit Testing in Visual Studio	190
10.9	Examples of Assertions	192

ie business school

#1 EUROPEAN BUSINESS SCHOOL
FINANCIAL TIMES 2013

#gobeyond

MASTER IN MANAGEMENT

Because achieving your dreams is your greatest challenge. IE Business School's Master in Management taught in English, Spanish or bilingually, trains young high performance professionals at the beginning of their career through an innovative and stimulating program that will help them reach their full potential.

- Choose your area of specialization.
- Customize your master through the different options offered.
- Global Immersion Weeks in locations such as London, Silicon Valley or Shanghai.

Because you change, we change with you.

www.ie.edu/master-management | mim.admissions@ie.edu |

10.10	Several Test Examples	193
10.11	Running Tests	199
10.12	Test Driven Development (TDD)	200
10.13	TDD Cycles	201
10.14	Claims for TDD	201
10.15	Summary	202
11	Case Study	203
11.1	The Problem	204
11.2	Preliminary Analysis	205
11.3	Further Analysis	211
11.4	Documenting the design using UML	216
11.5	Prototyping the Interface	220
11.6	Revising the Design to Accommodate Changing Requirements	221
11.7	Packaging the Classes	224
11.8	Programming the Message Classes	226
11.9	Programming the Client Classes	233
11.10	Creating and Handling UnknownClientException	235
11.11	Programming the Interface	237
11.12	Using Test Driven Development and Extending the System	241
11.13	Generating the Documentation	247
11.14	The Finished System	251
11.15	Running the System	252
11.6	Conclusions	254