Itä-Suomen ICT-polku

COURSE DATA

BASIC INFO

Name	Programming III - Object Oriented Programming		
Code	Savonia: ETA7300 Karelia: LTD6007 UEF: 3621412		
Name in Finnish	Ohjelmointi III – Olio-ohjelmointi		
Credits	5	**Grading**	0 - 5
Teaching period	1K		
Languge	Finnish		
Type	Savonia: compulsory course Karelia: compulsory course UEF/TKT: compulsory course		

Itä-Suomen ICT-polku

COURSE DESCRIPTION

Objectives	You understand the object-oriented programming paradigm
	You realize what a class is
	You are able to define a class
	You realize what an object is
	You are able to create instances of a class
	You realize what an attribute is
	You are able to create attributes into a class
	You realize what a method is
	You are able to create a dynamic method bound to an instance of a class
	You are able to create a static method
	You are able to use objects when modelling an application domain
	You are able to design classes according to an application domain model
	You realize what data encapsulation means
	You realize what is an accessor
	You realize what is an mutator
	You realize what a mutator is used for
	You realize what an accessor is used for
	You realize what access control attributes are
	You are able to restrict visibility of attributes
	You are able to restrict visibility of methods
	You are able to create accessors
	You realize what inheritance is about
	You are able to inherit a class
	You are able to override a method in a subclass
	You are able to restrict the visibility of inherited attributes in a subclass
	You are able to restrict the visibility of inherited methods in a subclass
	You realize what aggregation means
	You realize what composition means
	You realize what overriding a method means
	You are able to realize an aggregation
	You are able to realize an composition
	You are able to utilize code reusability in a larger project

	You realize the role of polymorphism in inheritance You realize the role of polymorphism in methods You realize what overriding a method means You realize how data type conversions work You are able to write an overridable method You are able to override a method You realize what exceptions are You realize how exceptions are used You are able to catch an exception You are able to throw an exception You are able to design an exception class of your own You realize what is an interface You are able to design interfaces You are able to realize interfaces You are able to realize several interfaces into one class You realize what is an abstract class You are able to design an abstract class You are able to design an abstract method You are able to realize an abstract method You are able to design an implementation model for the application domain You are able to choose inheritance, aggregation and composition regarding the needs of the application domain You realize what threads are You are able to use threads in your programs You realize parallel processing using threads You know the fundamental generic classes of the programming language You are able to utilize generic classes
Content	Object-oriented paradigm and modelling Classes and instances Data encapsulation Static and dynamic attributes and methods Accessors and mutators Inheritance Polymorphism

Itä-Suomen ICT-polku

	Abstract classes Interfaces Aggregation and composition Threads Exceptions
Modes of study	Pre-recorded theory lectures, demo sessions, exam.
Study materials	To be announced at the start of the course.
Teaching methods	Lectures 4h, videos 5h, demos 14h, exam 4h, independent work approx. 100h.
Prerequisites	Programming I - Basics of Programming Programming II – User Interface Programming
Other issues	