Course Content

Week	Content	Intended Learning Outcomes	Learning Activities
1	Welcome students to the course. Introduction to course outcomes and object oriented programming.	Students can identify the outcomes and overall content of the course	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks)
	Introduction of teachers and teaching assistants	Students can name the teaching team and their area of responsibility on the course	
	Introduction of learnIT	Students can log on to the learning platforms (learnIT and Adobe Connect Pro)	
	Introduction of students to each other	Create a profile in learnIT and upload a post in the welcome forum introducing themselves to each other. Students must interact with each other. Students can perform simple navigation and postings in the discussion forum.	
	Introduction to Java, OOP and the Java virtual machine	I students can recognize the terms of is OOP and Java	
	Introduction to NetBeans	students can recognize the term NetBeans	
	Introduction to object and classes		
2	Object and classes	Describe what is an object and a classes	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Object relations	Describe the common relationships between objects in OOP	
	Adapter Pattern	Describe the adapter pattern	
	Reference, copy, comparison and cloning	Describe what a reference is and apply it to different operations	
	Coupling, cohesion and encapsulation	Describe the concepts of coupling, cohesion and	

		encapsulation	
3	Nested classes	Define nested classes	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Inheritance	Define, describe, and apply inheritance	
	Abstract classes	Define nested classes	
	Interfaces	Define, describe, and apply interfaces	
4	Polymorphism	Define, describe, and apply polymorphism	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Design by contract	Describe the concept of design by contract in OOP	
	Singleton Pattern	Describe the singleton pattern	
	Immutable classes	Define and apply immutable classes Compare the different types of classes and apply them in the correct context	
	Builder Pattern and Factory Pattern	Describe the factory pattern and the builder pattern	
5	Dealing with exceptions	Define, describe, and apply exceptions	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Checked and Unchecked exceptions	Describe checked and unchecked exceptions	
	Assertions	Describe and apply assertions	
6	Abstract data types and containers	Define abstract data types and containers	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home
	Java Collection Framework	Apply abstract data types in Java	
	List, sets and maps	Describe and apply the most common container classes	
	Linked data	Define, describe, and apply data linking	

			assignments in group
7	Iterator pattern	Describe the iterator pattern	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Queue, stack and other collections	Describe and apply other container classes	
8	Generic types	Define and describe generic types	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Template pattern	Describe the template pattern	
9	Limitations of generics in Java	Apply generic types	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Sorting collections	Apply generics to collections	
	Comparator and comparable interface	Describe some common interfaces for comparing objects	
10	Threads	Define and describe threads	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Predictability and data consistency	Apply threads	
	Locks	Define and describe mutex lock	
11	Thread communication	Apply threads	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home
	Producer-consumer patters	Describe the producer-consumer pattern	
	I/O Streams	Define and describe I/O streams	

			assignments in group
12	Decorator patter	Describe the decorator pattern	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Serialization	Define, describe, and apply serialization	
	File and file chooser	Describe the File class	
13	Introduction to GUI Swings	Define, describe, and apply GUIs	1. Lecture(s) (several video clips) 2. Coding example(s) 3. Weekly synch Chat 4. Home assignment (a set of tasks) 5. Student group work required to solve home assignments in group
	Observer pattern	Describe the observer pattern	
	Event driven programming	Describe and apply event driven programming	