1.	Course title	Object-oriented programming		
2.	Course code	CSES201		
3.	Study program	Computer Science and Engineering, Computer Networks Technologies, Applied E-Technologies Education of Informatics, Informatics and Computer Engineering, Professional Informatics Studies, Academic Informatics Studies, Professional Information Technologies Studies		
4.	Unit offering the course	**FCSE**		
5.	Undergraduate/postgraduate/PhD	**Undergraduate**		
6.	Year/semester	7. ECTS: **6**		
8.	Teacher(s)	prof. dr. Dragan Mihajlov, prof. dr. Suzana Loshkovska, assoc. prof. dr. Ana Madevska-Bogdanova, assoc. prof. dr. Dejan Gjorgjevikj, assist. prof. dr. Nevena Ackovska, assist. prof. dr. Ivan Chorbev, assist. prof. dr. Ivica Dimitrovski, assist. prof. dr. Gjorgji Madzarov		
9.	Course prerequisites	None		
10.	Goals (competences): The goal of the course is to acquaint the student with the basic concepts of object-oriented programming. Therefore, the concepts of classes and objects will be introduced, encapsulation, inheritance and polymorphism. The students will be introduces to the concept of hierarchy of classes. Comparisons will be made of the implementation of the object-oriented concepts in different programming languages. After the completion of the course, the student will understand the principles of object-oriented programming and will be able to develop programs based on these concepts and principles.			
11.	Course content: Programming paradigms. Definition and basis of object-oriented programming. Characteristics of OO programming languages. Terminology (objects, classes, methods, encapsulation, abstraction, inheritance, polymorphism). Implementation of methods. Constructors. Destructors. Constructor overloading. Objects as arguments. Embedding objects. Inheritance. Multiple inheritances. Conversions among basic and derived classes. Overloading and redefining functions in derived classes. Polymorphism. Abstract classes. Interfaces. Exceptions.			
12.	Teaching methods: Lectures supported by presentations with slides, interactive lectures, exercises (use of equipment and software packages), real life examples, invited guest lecturers, preparation and defence of a project work and seminar thesis, learning in an e-environment (forums, consultations).			
13.	Total available time	6 ECTS x 30 hours = 180 hours		
14.	Distribution of the available time	30 + 45 + 30 + 75 = 180 hours		
15.	Teaching activities	15.1.	Lectures	30 hours
		15.2.	Training (labs, problem solving), seminar and team work	60 hours

16.	Other activities	16.1.	Project work	30 hours
		16.2.	Self study	30 hours
		16.3.	Home work	30 hours
17.	Grading			
	17.1.	Tests		75 points
	17.2.	Seminar work/project (written or oral presentation)		15 points
	17.3.	Active participation		10 points
18.	Grading criteria		to 50 points	5 (five) (F)
			from 51 to 60 points	6 (six) (E)
			from 61 to 70 points	7 (seven) (D)
			from 71 to 80 points	8 (eight) (C)
			from 81 to 90 points	9 (nine) (B)
			from 91 to 100 points	10 (ten) (A)
19.	Final exam prerequisites		Completed activities 15.1 and 15.2	
20.	Course language		Macedonian and English	
21.	Quality assurance methods		Internal evaluation and satisfaction polls	

22.	Literature					
	22.1.	Compulsory				
		No.	Authors	Title	Publisher	Year
		1.	Prata S.	C++ Primer Plus	the Waite Group	1998
		2.	Bruce Eckel	Thinking in Java, 4th edition	MindView	2006
		3.	Stroustrup B	The C++ Programming Language	AddisonWesley,	1997
	22.2.	Additional				
		No.	Authors	Title	Publisher	Year
		1.	Deitel, Deitel	How to program, Java, 8th edition	Prentice Hall	2010
		2.				
		3.				