2018.gada 21.maija rīkojuma Nr.196 – s
4.PIELIKUMS

VIDZEMES AUGSTSKOLA

FACULTY OF ENGINEERING
STUDY COURSE DESCRIPTION

Course Title:	**PYTHON OOP AND MODELLING**				
Course code (VAIS):					
Study programme:	**Information technologies**				
Level of Study programme:	☐	1st level professional higher education			
	☒	Professional Bachelor			
	☐	Professional Master			
	☐	PhD level			
Type of Study programme:	☒	Compulsory course (Part A)			
	☐	Professional specialization courses (Part B, compulsory)			
	☐	Professional specialization optional courses (Part B, optional)			
	☐	Elective courses (Part C)			
Course Workload:	**Credits**	**ECTS**	**Academic hours**	**Contact hours**	**Independent work hours**
FT (in LV: **PL)**	2	3	80	32	48
PT (in LV: **NL)**	2	3	80	10	70
Course Author/ Tutor:	**Kaspars Osis**				
	Assoc. Prof., Dr.sc.ing.				
	kaspars.osis@va.lv				
	Consultation: according to the schedule for each semester or per individual agreement.				
Course Form:	Full time (FT), Part time (PT)				
Study year, semester:	2nd year, 1st semester				
Language:	Latvian, English				
Prerequisites for the Course:	Basic knowledge and experience in programming languages – preferably Java and as a prerequisite in Python programming language (study course: Introduction to Python programming and data exploration); knowledge / insight about development of information systems.				
Course Summary:	The study course provides knowledge about Python object-oriented programming (OOP) and its application in applied Python OOP solutions. In the same time there are acquired knowledge and understanding about systems functional and structural analysis based on gained skills in development and analysis of UML diagrams.				
Course Methods:	Lectures, practical activities, group work, theory test, final assessment (project work assignment) etc.				
Assessment:	Examination (project work assignment)				
Requirements for Credits:	1. Successful completion of workshops/practical work assignments (at least 60% points of totally available). 2. Passed theoretical test. 3. Successful completion of project work assignment (at least 65% points of totally available). Final assessment consists of: workshops/practical work assignments, group work evaluations; theoretical test; project work assignment and project work assignment presentation. All practical work assignments have to be accepted (i.e. at least with 60% evaluation) in order to get the final evaluation within this course. 200 points system is used to come up with final evaluation. Table below lists totally available points for each activity.				

Work assignment or activity	**Points**
Practical work assignments	75
Theoretical test	20
Participation in class work activities	10
Project work assignment (exam)	80
Project work assignment presentation (exam)	15
Total	**200**

Final course evaluation (mark) calculation based on 200 points system is done as it follows below:

>= 93% (186-points) = 10	>= 75% (150-points) = 6
>= 90% (180-points) = 9	>= 70% (140-points) = 5
>= 85% (170-points) = 8	>= 65% (130-points) = 4
>= 80% (160-points) = 7	< 65% (130-points) = 3

Missing practical work assignment deadline: each missed day counts for subtraction of 5% from totally available points. It is required to acquire at least 60% from totally available points (not counting potential delay) in order to accept practical work assignment as done. There is provided a template which must be used for documenting practical work assignments – otherwise practical work assignment is not accepted for evaluation.

Abiding by the Academic Ethics

Students must abide by the academic and research ethics, Vidzeme University of Applied Sciences Ethics Regulations, incl.:

- study papers must be independently developed;
- the study work should reference all statements, ideas and data used that have been authored by someone else;
- appropriate data acquisition methods should be used in the acquisition of data, the research ethics must be respected, empirical data must be collected independently and cannot be distorted or falsified;
- the examination must be carried out by the student independently, without the use of supporting materials and/or consultations with other students, unless the lecturer states otherwise.

In the event of non-compliance with the academic and research ethics, punishment is imposed in accordance with the ViA Ethics Regulations and the study course must be re-taken, unless the punishment is extramarital.

Learning Outcomes; the evaluation methods and criteria

Learning Outcomes	**The evaluation methods and criteria**
Knowledge	
Knowledge on UML application necessity and accordant cases.	Development of particular UML solution. Passed theoretical test.
Knowledge about UML diagram types, concrete diagrams, and modelling based on them.	Development of particular UML solution. Passed theoretical test.
Knowledge about object-based and object-oriented programming in perspective of Python.	Development of Python solution. Passed theoretical test.
Knowledge about event-driven and multithreaded programming in context of Python.	Passed theoretical test.
Skills	
To develop UML based solution from use-case perspective.	Developed practical group work.
To develop UML and Python based solutions from the structural perspective.	Developed practical group work.

	To develop UML and Python based solutions from the structural perspective with increased level of details.	Developed practical group work.
	To develop introductory level UML and Python based solutions from behaviour concept perspective.	Developed practical group work.
	Competency	
	Use correct UML and Python solutions terminology. To choose appropriate technological approaches for particular assignment implementation.	Course project development and presentation.
	Independently perform UML and Python solutions design and development.	Course project development and presentation.
	To solve UML and Python solutions basic issues.	Course project development and presentation.
Course Compulsory literature:	1. Miller, B., N., Anderson, J., Ranum, D., L. Python Programming In Context, 3rd ed., Jones and Bartlett Publishers, Burlington, MA, 2019. 2. Seidl, M., Scholz, M., Huemer, Ch., Kappel, G. UML @ Classroom: An Introduction to Object-Oriented Modeling, Springer International Publishing, Heidelberg, 2015. 3. Philips, D. Python 3 Object-Oriented Programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8, 3rd ed., Packt Publishing, Birmingham, 2018.	
Course additional literature:	1. Fowler, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd edition, Addison Wesley, 2004. 2. Deitel, P. Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and The Cloud, Global Edition, Pearson Education, 2021. 3. Osis, J., Donins, U. Topological UML Modeling: an improved approach for domain modeling and software development, 1st ed., Elsevier, Cambridge, MA, 2017. 4. Steinpichler, D., Kargl, H. Project Management with UML and Enterprise Architect, 8th ed., SparxSystems Software GmbH, Vienna, 2011. 5. Miles, R., Hamilton, K. *Learning UML 2.0*, 1st edition, O'Reilly Media, 2006.	
Course confirmation date:		
Date of course description update:		

Study Course Plan for Full Time Students:

Date	Theme	Academic hours		Study Form/ Organization of independent work of students and task description
		Contact hours	Independent work hours	
	Introduction. Beginnings and basics of UML. Overview of UML editing tools. MDA and its relation with UML.	4	4	Theoretical lecture. Several topics covering practical work. Group work.
	Use Case diagram, components, development principles. Documenting use cases.	4	4	Theoretical lecture. Several topics covering practical work. Group work.

	UML structure diagram, modelling. Python object-based programming. Python objects.	4	4	Theoretical lecture. Several topics covering practical work. Group work.
	UML sequence diagram. Python: object-oriented design, construction of classes.	4	7	Theoretical lecture. Several topics covering practical work. Group work.
	UML state machine diagram. Using Python objects in simulation.	4	6	Theoretical lecture. Several topics covering practical work. Group work.
	UML activity diagram. Python: inheritance, polymorphism, graphics library. LPW.	4	5	Theoretical lecture. Several topics covering practical work. Group work.
	Re-engineering. Python and reengineering. Python: event-driven programming, multithreading, event handlers, static variables. Creating simple video game.	4	2	Theoretical lecture. Group work.
	Final examination.	4	24	Course project development and presentation.
	Hours total:	**32**	**48**	

Note: lecturer keeps the rights to make changes in the course plan.

Study Course Plan for Part Time Students:

Date	**Theme**	**Academic hours**		**Study Form/ Organization of independent work of students and task description**
		Contact hours	**Independent work hours**	
	Introduction. Beginnings and basics of UML. Overview of UML editing tools. MDA and its relation with UML. Use Case diagram, components, development principles. Documenting use cases.	2	15	Theoretical lecture. Several topics covering practical work. Group work.
	UML structure diagram, modelling. Python object-based programming. Python objects. UML sequence diagram. Python: object-oriented design, construction of classes.	2	15	Theoretical lecture. Several topics covering practical work. Group work

	UML state machine diagram. Using Python objects in simulation. UML activity diagram. Python: inheritance, polymorphism, graphics library. LPW.	2	15	Theoretical lecture. Several topics covering practical work. Group work
	Python: event-driven programming, multithreading, event handlers, static variables. Creating simple video game.	2	3	Theoretical lecture. Group work
	Final examination.	2	22	Course project development and presentation.
	Hours total:	**10**	**70**	

Note: lecturer keeps the rights to make changes in the course plan.