University of Gdańsk

Subject card

Subject name and code	Automatic Testing, PG_00204168						
Field of study	Informatics						
Date of commencement of studies	October 2026		Academic year of realisation of subject		2027/2028		
Education level	Bachelor's studies		Subject group		Obligatory subject group in the field of study Subject group related to practical vocational preparation		
Mode of study	full-time studies		Mode of delivery		at the university		
Year of study	2		Language of instruction		Polish		
Semester of study	3		ECTS credits		3.0		
Learning profile	practical		Assessment form		credit		
Conducting unit							
Name and surname of lecturer (lecturers)	Subject supervisor		dr Tomasz Borzyszkowski				
	Teachers						
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	30.0	0.0	0.0	45
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours		Self-study	SUM
	Number of study hours	45		0.0		30.0	75
Subject objectives	Familiarisation with modern methods of quality assurance in IT projects through software testing and validation. Understand the requirements for IT systems (e.g. availability, performance) and the tools supporting automated software testing. To become familiar with the practices used in the automation of various types of tests (e.g. unit, integration and user interface tests).						

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFPL3_U04] is able to use the acquired knowledge when creating, running and testing programs using dedicated tools and design patterns	use advanced functionalities of operating systems, in particular related to networking aspects, virtualisation, containerisation and other cloud technologies	[SU2] presentation/project/paper/ report
[INFPL3_K02] is ready to recognize the importance of knowledge in solving cognitive problems and practical and seeking opinions experts in case of difficulties with independent problem solving	is able to formulate precise questions in order to deepen his/ her own understanding of a given topic or to find missing pieces of reasoning, is ready to use technical documentation, professional literature and expert sources	[SK8] observation of student's independent or team work
[INFPL3_W08] knows and understands facts and methods to an advanced degree in the field of the use of software development, maintenance and test tools and environments; apply this knowledge to create efficient, scalable and secure applications	has knowledge in the use of tools and environments for software development, testing and maintenance	[SW2] presentation/project/paper/ report
[INFPL3_W07] knows and understands facts and methods to an advanced degree in the field of designing, developing, testing, implementing and maintaining web applications and their security; applies this knowledge in practical projects, creating web applications and preparing their functional and performance tests	has knowledge of the design, development, testing, implementation and maintenance of web applications and their security	[SW2] presentation/project/paper/ report

Subject contents

Methodologies and types of software tests (test pyramid)Software test planning as part of the software development cycle (formulation of test scenarios)Test automation: - automation tools - automated testing as a means of quality assurance in the software development cycle - integration of automated testing into the implementation process

Prerequisites and co-requisites

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Project presentation, observation of student work	51.0%	90.0%
Activity in class	51.0%	10.0%

Recommended reading

Basic literature	1 Beck Kent, TDD. The art of creating good code Helion Publishing, 2020.2 Harry J.W. Percival. TDD in practice. Reliable code in Python. Helion Publishing, 2020.
Supplementary literature	None
eResources addresses	

Example issues/ example questions/ tasks being completed

1. Using TDD methodology, write a class/function called hamming that calculates the Hamming distance for given test data.2. For given requirements, scenarios and use cases, design and implement behavioural tests using the behave library in Python.

Work placement

Not applicable

Document generated electronically. Does not require a seal or signature.