University of Gdańsk

Subject card

| | | | |
|---|---|---|---|
| Subject name and code | Automatic testing (P), PG_00143998 | | |
| Field of study | Informatics | | |
| Date of commencement of studies | October 2025 | Academic year of realisation of subject | 2026/2027 |
| Education level | Bachelor's studies | Subject group | Obligatory subject group in the field of study<br>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 |
|---|---|---|
| [INFL3_U04] can create, run and test 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 |
| [INFL3_W07] has knowledge of designing, developing, testing, implementing and maintaining web applications and their security | has knowledge in the use of tools and environments for software development, testing and maintenance | [SW2] presentation/project/paper/ report |
| [INFL3_W08] has knowledge of the use of software development, maintenance and test tools and environments | has knowledge of the design, development, testing, implementation and maintenance of web applications and their security | [SW2] presentation/project/paper/ report |
| [INFL3_K02] can precisely formulate questions to deepen his/ her own understanding of a given topic or to find missing elements of reasoning | 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 | [SK8] observation of student's independent or team work |

<table>
<tr><td>Subject contents</td><td>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</td></tr>
<tr><td>Prerequisites and co-requisites</td><td>None</td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>Activity in class</td><td>0.0%</td><td>10.0%</td></tr>
<tr><td>Project presentation, observation of student work</td><td>51.0%</td><td>90.0%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>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.</td></tr>
<tr><td>Supplementary literature</td><td>None</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>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.</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

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