AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Module name: Introduction to Java Programming

Academic year: 2019/2020 Code: IETE-1-606-s ECTS credits: 3

Faculty of: Computer Science, Electronics and Telecommunications

Field of study: Electronics and Telecommunications Specialty: —

Study level: First-cycle studies Form and type of study: Full-time studies

Lecture language: English Profile of education: Academic (A) Semester: 6

Course homepage: —

Responsible teacher: dr inż. Orzechowski Tomasz Marcin (tomeko@agh.edu.pl)

Module summary

This module is intended to provide students who do not already have significant computing experience, with the ability to use Java as their programming tool for their current & future projects.

Description of learning outcomes for module

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Connections with FLO	Method of learning outcomes verification (form of completion)
Skills: he can			
M_U001	Students can implement Java applications with different levels of difficulties, using different Java packages.	ETE1A_U15	Project
M_U002	The student practically uses the knowledge of the stages of software development.	ETE1A_U02, ETE1A_U04	Execution of a project, Project
Knowledge: he knows and understands			
M_W001	Student has detailed knowledge about software engineering – analysis and designing software projects	ETE1A_W06	Activity during classes, Execution of a project, Oral answer
M_W002	Student knows and understands basic elements and mechanisms of object oriented programming language	ETE1A_W06	Participation in a discussion, Activity during classes, Execution of a project, Project

Number of hours for each form of classes

Suma	Form of classes										
	Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Prace kontrolne i przejściowe	Lektorat
36	12	0	0	24	0	0	0	0	0	0	0

FLO matrix in relation to forms of classes

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Form of classes										
		Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Prace kontrolne i przejściowe	Lektorat
Skills: he can												
M_U001	Students can implement Java applications with different levels of difficulties, using different Java packages.	-	-	-	+	-	-	-	-	-	-	-
M_U002	The student practically uses the knowledge of the stages of software development.	-	-	-	-	-	-	-	-	-	-	-
Knowledge: he knows and understands												
M_W001	Student has detailed knowledge about software engineering – analysis and designing software projects	-	-	-	-	-	-	-	-	-	-	-
M_W002	Student knows and understands basic elements and mechanisms of object oriented programming language	+	-	-	-	-	-	-	-	-	-	-

Student workload (ECTS credits balance)

Student activity form	Student workload
Udział w zajęciach dydaktycznych/praktyka	36 h
Realization of independently performed tasks	39 h
Summary student workload	75 h
Module ECTS credits	3 ECTS

Additional information

Module content

Lectures

Organizing classes (1h)
- discussing the rules and validation
- important information about individual projects and terms

Java and other programming languages (2h)
- Basic definitions and notations;
- Differences between Java and C++ including: data types
- Syntax of Java: keywords, operators, statements
- Documentation and Tutorials
- JDK / JRE – availabity

NetBeans and Eclipse IDE (1h)
- availability
- installation
- configuration

Exceptions and Streams (2h)
- Throwable, Error and Exceprion classes
- Input / output operations, data streams types

Collections in Java (1h)
- The use of lists and sets
- Vector and ArrayList – differences and similarities
- sorting

Java SWING (2h)
- containers and components
- layouts
- events handling

log4J (1h)
- logging mechanism in Java

JDBC (2h)
- SQL in Java
- the use of mySQL and SQLITE

JSoup (1h)
- parsing data from Internet
- installation and configuration

Threads (1h)
- Threads and Pools – differences and similarities

Project classes

Selection the topic to be implemented
- detailed discussion about chosen task.
- the work plan and the relevant elements to be evaluated.

analysis phase
student presentation:
- understanding the essence of the problem;
- analysis of selected tools

design phase
student presentation:
- system structure,
- components and relationships between components,
- interfaces

implementation phase and validation process
- demonstration of alpha and beta versions of the application
- testing the application with an intention of finding bugs
- improved the application performance

final evaluation
- final evaluation of practical work

Teaching methods and techniques:
Lectures: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pokazy odnoszące się do prezentowanych zagadnień.
Project classes: Studenci wykonują zadany projekt samodzielnie, bez większej ingerencji prowadzącego. Ma to wykształcić poczucie odpowiedzialności za pracę w grupie oraz odpowiedzialności za podejmowane decyzje.

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:
Nie określono

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:
Lectures:
– Attendance is mandatory: No
– Participation rules in classes: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z syllabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.
Project classes:
– Attendance is mandatory: Yes
– Participation rules in classes: Studenci wykonują prace praktyczne mające na celu uzyskanie kompetencji zakładanych przez syllabus. Ocenie podlega sposób wykonania projektu oraz efekt końcowy.

Method of calculating the final grade
Following elements are considered when the total grate is calculated: the analysis of the received programming problem, correct presented project of the application, presentations of alpha and beta versions of the application, validation of the application by carrying out appropriate tests.

Positive evaluation can only be obtained when the final result is a properly functioning application that is accompanied by appropriate documentation.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:
Nie określono

Prerequisites and additional requirements
Students are expected to have some previous programming experience (not necessarily with an object-

oriented language). It is possible to take the course without any previous experience, but this will almost certainly be challenging, and require additional time.

Recommended literature and teaching resources

1. Cay S. Horstmann and Gary Cornell, Core Java I – Fundamentals, ISBN-10: 0137081898
2. Cay S. Horstmann and Gary Cornell, Core Java II – Advanced Features, ISBN-10: 0134177290
3. Online documentation: Java 8 API: https://docs.oracle.com/javase/8/docs/api
4. Java Tutorial: https://docs.oracle.com/javase/tutorial

Scientific publications of module course instructors related to the topic of the module

Additional scientific publications not specified

Additional information

None