Bialystok University of Technology									
Field of study	**Computer Science**							**Degree level and programme type**	**Engineer's degree full-time programme**
Specialization/ diploma path	**---**							**Study profile**	academic
Course name	**Advanced Object-Oriented Programming Techniques**							**Course code**	**FCS-00090**
								Course type	obligatory
Forms and number of hours of tuition	L	C	LC	P	SW	FW	S	**Semester**	1
	30				30			**No. of ECTS credits**	**6**
Entry requirements	Object Oriented Programming (FCS-00012),								
Course objectives	Introduction to selected design patterns and their example applications in simple programs written and refactored during the Sw.								
Course content	Classes: Notion of the design pattern. Classification of the design patterns. Patterns: Singleton, Factory Method, Prototype, Abstract Factory, Builder, Proxy, Adapter, Decorator, Composite, Flyweight, Facade, Bridge, Command, Strategy, Template Method, Iterator, State, Mediator, Observer, Visitor. Specialization workshop: Writing/refactoring short program using chosen design patterns.								
Teaching methods	informative lecture, lecture problem, programming,								
Assessment method	Lecture: Final test covering the design patterns. Specialization workshop: short programs implementing the design patterns, written/refactored in Java by students, presented and explained orally to the teacher								

Symbol of learning outcome	**Learning outcomes**	**Reference to the learning outcomes for the field of study**
LO1	enumerates and describes design patterns (including examples in written programs)	K_W06
LO2	uses the design patterns in simple programs	K_U04
LO3	designs (using the design patterns) and implements simple information systems	K_U06
LO4	acquires extra knowledge concerning detail mechanisms of the selected programming languages from their technical documentation	K_U06 K_K01

Symbol of learning outcome	**Methods of assessing the learning outcomes**	**Type of tuition during which the outcome is assessed**
LO1	L: written or oral evaluation Sw: short teoretical tests	L, Sw
LO2	evaluation of written programs	Sw
LO3	Evaluation of the simple programs: their design and implementations	Sw
LO4	Evaluations of the simple programs (concernig used API of the programming language)	Sw

Student workload (in hours)	**No. of hours**
Calculation	
1 - Attendance at lectures -	30
2 - Attendance at specialistic workshops -	30
3 - Preparation for specialization workshops (from the lecture and other sources) -	30
4 - Participation in student-teacher sessions -	5
5 - Realization of project tasks (with presentation) -	45
6 - Preparation for test -	10
TOTAL:	**150**

Quantitative indicators	**HOURS**	**No. of ECTS credits**
Student workload – activities that require direct teacher participation	65 (4)+(2)+(1)	2.6
Student workload – practical activities	105 (5)+(3)+(2)	4.2

Basic references	1. Gamma, E., Helm, R., Johnson, R., Vlissides,J.: Design Patterns: elements of reusable object-oriented software. Addison-Wesley Publ., 1995. 2. Buschmann, F.: Pattern-oriented software architecture. [Vol. 1], A system of patterns. John Wiley & Sons, 2008. 3. Buschmann, F., Henney, K., Schmidt, D.C.: Pattern-oriented software architecture. Vol.5, On patterns and pattern languages. John Wiley a. Sons, 2007.
Supplementary references	1. Vaskaran Sarcar: Java Design Patterns. A Hands-On Experience with Real-World Examples.Apress, Berkeley, CA, 2019 2. Kamalmeet Singh, Adrian Ianculescu, Lucian-Paul Torje: Design patterns and best practices in Java : a comprehensive guide to building smart and reusable code in Java. Packt Publishing, Limited, 2018 3. Wessel Badenhorst: Practical Python Design Patterns: Pythonic Solutions to Common Problems. Apress, Berkeley, CA, 2017.

Organisational unit conducting the course	Software Department	**Date of issuing the programme**
Author of the programme	dr inż. Cezary Bołdak	Feb. 17, 2022

L – lecture, C – classes, LC – laboratory classes, P – project, SW – specialization workshop, FW – field work,S – seminar