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	**Software Development Tools**	**Course code**	**FCS-00071**
		Course type	obligatory

Forms and number of hours of tuition	L	C	LC	P	SW	FW	S	**Semester**	2
	15				15			**No. of ECTS credits**	**6**

Entry requirements	Programming Basics (FCS-00031),
Course objectives	The purpose of the lecture is to present engineering methods and tools that support the software development process. These methods and tools consider different stages of software development from requirements gathering, through implementation and testing. The purpose of the specialization workshop is to practically present tools that support the process of software development from the moment of defining requirements, project implementation and deployment.
Course content	Lecture: Joel's test, integrated development environments (IDEs), code and changes management (software versioning and revision control systems), dynamic (run-time) software testing, software profiling/software performance testing, source code documentation, functional tests, bug management, requirement management, GUI prototyping tools, software distribution - instalators Specialization workshop: integrated development environments (IDEs) (e.g., MS VS, Eclipse, NetBeans, Jupyter), revision control systems (SVN, GIT), debugging, run-time software testing in unmanaged and managed code, time and memory profiling, source code documention, bug manangement, requirement management, GUI prototyping tools, software distribution - instalators
Teaching methods	lecture problem, programming,
Assessment method	Lectures: written examination, Practical classes: two written tests.

Symbol of learning outcome	**Learning outcomes**	**Reference to the learning outcomes for the field of study**
LO1	Knows the selected tools using to specify, design, develop and testing software applications.	K_W06
LO2	Knows and understands the role of software engineering tools. Knows the categories of tools supporting software application design. Has basic knowledge about the current tools supporting software delopment.	K_W06
LO3	Is able to choose and use properly integrated development tools, revision control systems and tools for run time testing of software applications. Is able to compare this kind of software development tools in accordance with their usability and enonomic issues (speed, cost, functionality).	K_U06
LO4	Is able to choose and use properly tools for software profiling, documentation generation, use case testing, bug management and distribution. Is able to compare this kind of tools in accordance with their usability and enonomic issues (speed, cost, functionality).	K_U06

Symbol of learning outcome	**Methods of assessing the learning outcomes**	**Type of tuition during which the outcome is assessed**
LO1	written examination	L
LO2	written examination	L
LO3	first written test, work during classes	Sw
LO4	second written test, work during classes	Sw

	Student workload (in hours)	**No. of hours**
Calculation		
	1 - Attendance at lectures -	15
	2 - Attendance at classes -	15
	3 - Report preparation and homeworks -	75
	4 - Participation in student-teacher sessions -	5
	5 - Preparation for the exam -	20
	6 - Preparation for classes -	20
	TOTAL:	**150**

Quantitative indicators	**HOURS**	**No. of ECTS credits**
Student workload – activities that require direct teacher participation	55 (4)+(2)+(1)+(5)	2.2
Student workload – practical activities	110 (2)+(3)+(6)	4.4

Basic references	1. B. Collins-Sussman, B.W. Fitzpatrick, C.M. Pilato, Version Control with Subversion, http://svnbook.red-bean.com/en/1.7/svn-book.pdf 2. J. Spolsky, The Best Software Writing, Apress, 2005 3. Microsoft, Patterns & Practices: Performance Testing Guidance for Web Applications, 2007, https://perftestingguide.codeplex.com 4. I. Sommerville, Software engineering, Pearson Education, Boston, 2004. 5. P. Glavich, C. Farrell, .NET Performance Testing and Optimization The Complete Guide, Simple Talk Publishing, 2010	
Supplementary references	1. H. van Vliet, Software engineering :principles and practice, John Wiley and Sons, 2008. 2. D. Spinellis, Code Reading: The Open Source Perspective, Addison-Wesley Professional, 2003 3. M. Fowler, K. Beck, D. Roberts, E. Gamma, Refactoring, Improving the Design of Existing Code, Addison-Wesley Professional, 1999 4. G. J. Myers, C. Sandler, T. Badgett, The Art of Software Testing Hardcover, Wiley, 2011	
Organisational unit conducting the course	Software Department	**Date of issuing the programme**
Author of the programme	dr inż. Krzysztof Jurczuk	Feb. 17, 2022

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