COURSE DESCRIPTION

TECHSummer 2020

Course name: **Introduction to DevOps and automation**

The course's form	***Lecture***	***Tutorial***	***Laboratory***	***Project***	***Seminar***
Total number of hours	-	-	*30*	-	-
Form of completion	-	-	*passing exercises*	-	-

- Initial requirements:

  Familiarity with any modern programming language is necessary.

- Name, surname, title of teachers:

  - Szymon Datko, MSc

- Course's aims and educational outcomes:

  Course's aims:

  The course aims to introduce the attenders with fundamental concepts of modern Development and Operations ideas and tools (commonly abbreviated as *DevOps*).

  Outcomes in terms of knowledge:

  - knowledge how version control systems work and track changes,
  - familiarity with fundamental concepts of infrastructure as a code approaches,
  - knowledge about common application development processes driven by tests.

  Outcomes in terms of skills:

  - ability to navigate through git tree and commit changes from CLI,
  - skill to develop a simple programming project in Python from scratch,
  - ability to build application container and start containerized application.

  Outcomes in terms of competences:

  - understanding the importance of version control systems nowadays,
  - knowledge of good practices in application development using Python,
  - understanding what challenges the containers and other concepts help to overcome.

- Form of teaching (traditional / e-learning):

  traditional

- Short description of the course content:

  In a series of practical lessons, the attendees will learn the basic concepts of currently common practices for developing the software and operating it in various environments, which is often referred as *DevOps*. First, the students will learn how to use command line interface for navigating through the Linux environments and write Bash scripts to automate common tasks. They will also familiarize with GIT version control system – its key concepts and how to use it for tracking changes in their projects. Then, the fundaments of infrastructure as a code will be also introduced on the example of popular Ansible tool.

  The example of good practices in software developments will be presents on the example of Python programming language, involving such concepts as virtual environments and tests driven development. Next, it will be shown how today application can be delivered in system-independent manners, utilizing the concept of Linux containers. Finally, the idea of Continuous Integration and Continuous Delivery will be presented on the example of simple pipeline to implement.

  The lessons will consist of workshops with introductory lectures, guided exercises and self-work, so familiarity with any modern programming language is necessary. All the workshops will be conducted in the computer lab and will base on Ubuntu Linux 20.04 with Python 3.

- Laboratory – content:

Form of classes - laboratory		Number of hours
1	Introduction to course, dev environment and tools	2
2	Getting started with CLI, writing simple Bash scripts	4
3	Learning and understanding GIT version control system	4
4	Infrastructure as a code using Ansible	4
5	Developing a simple Python project with virtual environments	4
6	Introduction to unit testing and functional testing	4
7	Working with Docker, building containers	4
8	Preparing Continuous Integration pipeline	4

- Basic literature:

  1. Gene Kim, Jez Humble, Patrick Debois, John Willis, John Allspaw, *“The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations”*, IT Revolution Press, 2016.
     (ISBN 978-1942788003)
  2. Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin, *“UNIX and Linux System Administration Handbook”*, 5th edition, Addison-Wesley 2017.
     (ISBN 978-0134277554)

- Additional literature:

  1. Noah Gift, Kennedy Behrman, Alfredo Deza, Grig Gheorghiu, *“Python for DevOps: Learn Ruthlessly Effective Automation”*, O'Reilly Media, 2019.
     (ISBN 978-1492057697)
  2. Jez Humble, David Farley, *“Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation”*, Addison-Wesley, 2010.
     (ISBN: 978-0321601919)

- Completion rules:

Laboratory: completion of exercises

Each exercise: result from range < 2.0; 5.5 >

Final result = average of all results + activity bonuses/penalties

Final result	Grade
< 5.25; +∞ >	5.5
< 4.75; 5.25)	5.0
< 4.25; 4.75)	4.5
< 3.75; 4.25)	4.0
< 3.25; 3.75)	3.5
< 3.00; 3.25)	3.0
< -∞; 3.00)	2.0