

Course Syllabus**Course from study programme for the cycle: 2022/2023****I. General Information**

Course name	Internet applications development
Programme	Informatics
Level of studies (BA, BSc, MA, MSc, long-cycle MA)	BA
Form of studies (full-time, part-time)	Full-time
Discipline	Informatics
Language of instruction	English

Course coordinator	Rafał Stęgiński, PhD
--------------------	----------------------

Type of class (<i>use only the types mentioned below</i>)	Number of teaching hours	Semester	ECTS Points
lecture			5
tutorial	30	IV	
classes			
laboratory classes	30	IV	
workshops			
seminar			
introductory seminar			
foreign language classes			
practical placement			
field work			
diploma laboratory			
translation classes			
study visit			

Course pre-requisites	Basics of algorithms and programming
-----------------------	--------------------------------------

II. Course Objectives

C1 - Getting to know the basics of the PHP language
C2 - Getting acquainted with web application programming techniques
C3 - Familiarizing with the programming techniques of console applications
C4 - Getting to know the basic design patterns
C5 - Getting acquainted with the structure and the cycle of web application implementation

III. Course learning outcomes with reference to programme learning outcomes

Symbol	Description of course learning outcome	Reference to programme learning outcome
KNOWLEDGE		
W_01	Has general knowledge of algorithmics, design and programming, operating systems, computer networks, software engineering, databases, artificial intelligence and computer graphics	K_W06
SKILLS		
U_01	Is able to independently acquire and use information helpful in solving specific IT problems from technical documentation, help files as well as Internet resources and available literature	K_U02
U_02	Knows language which describes application development process	K_U04
SOCIAL COMPETENCIES		
K_01	Knows own limitations and knowledge which should be extended.	K_K01

IV. Course Content

1. The WWW network
 - a. The http protocol
 - b. GET and POST requests
 - c. Processing on the client's side
 - d. Processing on the server side
 - e. REST
2. Syntax of the EcmaScript and Typescript language
 - b. Constants, variables, expressions and operators
 - c. Control instructions
 - d. Functions
 - e. Encoding standards
3. Object-oriented programming
 - a. Classes and objects
 - b. Constructors, destructors and cloning
 - c. Components
 - d. Inheritance
 - e. Specifications of component visibility
 - f. Static components
 - g. Permanent
 - h. Abstract classes
 - i. Interfaces
 - j. NodeJS and libraries
4. Design patterns
 - a. Basic information about design patterns
 - b. Selected design patterns
5. ORM software
6. Software framework

V. Didactic methods used and forms of assessment of learning outcomes

Symbol	Didactic methods (choose from the list)	Forms of assessment (choose from the list)	Documentation type (choose from the list)
KNOWLEDGE			
W_01	Conversational lecture, Guided practice	Exam	Protocol
SKILLS			
U_01	Practical classes design thinking	Preparation / implementation of the project	Project rating card
U_02	Practical classes design thinking	Preparation / implementation of the project	Project rating card
SOCIAL COMPETENCIES			
K_01	Discussion design thinking	Observation	Protocol

VI. Grading criteria, weighting factors.....

At grade 3, the student can:

W1 - can characterize the differences between the interpretation and compilation of the discuss the syntax of the EcmaScript and TypeScript language

U1 - run sample internet applications made in various frameworks

U2 - implement simple applications based on processing strings, arrays and files

K1 - can formulate opinions on basic ES and TS language constructs

At grade 4, the student can:

W1 - contrastively discuss the syntax of the ES and TS language in relation to any other language (eg C ++), exchange and briefly characterize the known design patterns

U1 - implement object-oriented libraries that solve more advanced tasks

U2 - use your own libraries to implement the application

K1 - work individually and in groups to plan work on the application

At grade 5 the student can:

W1 - give examples of the use of the discussed design patterns

U1 - use design patterns in practice to implement your own libraries

U2 - publish your own libraries as Open Source projects

VII. Student workload

Form of activity	Number of hours
Number of contact hours (with the teacher)	90
Number of hours of individual student work	50

VIII. Literature

Basic literature
Shelley Powers, Learning Node. Moving to the Server-Side. 2nd Edition., O'Reilly David Flanagan, JavaScript The Definitive Guide: Master the World's Most-Used Programming Language, O'Reilly
Additional literature
https://nodejs.org/en/docs/