**General Information:**

**Name of Course: Programming 2**

**Course Code: IVB305ANMI**

**Semester:** 2nd

**Number of Credits: 4**

**Allotment of Hours per Week:** 2 lectures, 3 labours

**Evaluation: Mid-term**(with grade)

**Prerequisites: Programming 1**

**Instructors: Zoltan Zidarics**

Office: 7624 Hungary, Pécs, Boszorkány u. 2. Office No B233

E-mail: [z](mailto:perjesi@mik.pte.hu)amek@vili.pmf.hu

Office Phone: +36 72 503650/23851

**Java programming**

Programming in Java language in basic level. Upon completion of this course the student should be able to: **interpret,** and **put into practice**

1. using Java language in different operating systems,
2. using Eclipse as an integrated development environment,
3. strong knowledge of object oriented programming paradigm
4. basic activities:
   1. creating a project
   2. creating a test environment
   3. documenting codes
   4. XML and JSON knowledges
   5. using different build tools (Maven, Ant, Gradle)
   6. version controlling (Git, Gitlab)
   7. Java basics (Lists, Basic I/O, Database access, Graphical surface, Threads, Generics, Functional programming, Design patterns)

**General Course Description and Main Content:**

Brief Syllabus: This lecture and practical based course aims to give computer science engineering students a solid java basis through covering the following topics:

* Creating and managing Java projects
* Sharing codes in a version controlling system
* Working in a developer workgroup, managing software lifecycle

Students learn the basics of programming enabling them to interpret and understand engineering sciences and through solving elementary tasks they deepen their basic theoretical knowledge in the field of engineering. The practical sessions are designed to complement the requirements of different specialisations.

**Methodology:**

The presentations give an introduction to important part of Java language and software development techniques of exercise solving and the basic theory. Equal emphasis is given to learning how to construct and manage a Java project.

**Schedule:**

Study period in 15 weeks: February 5 - May 18 (2018)

1. Introduction of Java language. Why Java?
2. Object Oriented Paradigm and creating projects.
3. XML and JSON, Java build tools
4. Version controlling (**Homework I**)
5. Using Lists (**Homework II**)
6. Basic I/O (**Homework III**)
7. Database Management
8. Graphical surface
9. *Spring break*
10. Threads in Java
11. **Generics in Java**
12. **Test 1**
13. Functional programming
14. Design patterns
15. **Test 2**

Correction period: May 22-24 (2018)

Exam: The subject ends with mid-term grade

**Attendance:**

Attending is not required all classes, but it needs all homeworks and project work at least 65% level.

**Evaluation + Grading**

Grading will follow the course structure with the following weight:

1. tests 50%
2. Homeworks 20 %.
3. project work 30 %
4. Offered exam grade: over 55 % during the study and correction period.
5. Written project work at the begin of exam period. A minimum of 55% is required to pass the project work.

**Grading scale**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Numeric Grade: | 5 | 4 | 3 | 2 | 1 |
| Evaluation in points: | 89%-100% | 77%-88% | 66%-76% | 55%-65% | 0-54% |

**Students with special needs:**

Students with special physical needs and requiring special assistance must first register with the Dean of the Students Office. All reasonable requests to provide an equal learning environment for all students is to be assured.

**Required Reading and other Materials will be equivalent to:**

**The Java tutorials** [**https://docs.oracle.com/javase/tutorial/index.html**](https://docs.oracle.com/javase/tutorial/index.html)

**XML tutorial** [**https://www.w3schools.com/xml/default.asp**](https://www.w3schools.com/xml/default.asp)

**JSON tutorial** [**https://www.w3schools.com/js/js\_json\_intro.asp**](https://www.w3schools.com/js/js_json_intro.asp)

**Apache Ant manual** [**https://ant.apache.org/manual/**](https://ant.apache.org/manual/)

**APACHE maven manual** [**https://maven.apache.org/what-is-maven.html**](https://maven.apache.org/what-is-maven.html)

**GIT documentation** [**https://git-scm.com/doc**](https://git-scm.com/doc)

**gitlab documentation** [**https://docs.gitlab.com/**](https://docs.gitlab.com/)

**DESIGN PATTERNS https://en.wikipedia.org/wiki/Software\_design\_pattern**