**COURSE SYLLABUS**

***ADVANCED PROGRAMMING ELEMENTS***

**1. Program identification details**

|  |  |
| --- | --- |
| 1.1 Higher education institution | „Ovidius” University of Constanta |
| 1.2 Faculty | Faculty Mathematics and Computer Science |
| 1.3 Department | Mathematics and Computer Science |
| 1.4 Field of studies | **Computer Science** |
| 1.5 Cycle of studies (degree) | Master |
| 1.6 Degree program/qualification | **Cyber Security and Machine Learning** |
| 1.7 Academic year | **2022-2023** |

**2. Course identification details**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2.1 Course title | | | ADVANCED PROGRAMMING ELEMENTS | | | | |
| 2.2 Course code | | | FMI.CSML.I.1.03 | | | | |
| 2.3 Instructor | | | Prof.univ.dr. POPOVICI Dorin-Mircea | | | | |
| 2.4 Teaching assistant | | | Prof.univ.dr. POPOVICI Dorin-Mircea | | | | |
| 2.5 Year | I | 2.6 Semester | 1 | 2.7. Evaluation type | E | 2.8 Course type \*/\*\* | DAP/DI |

*\* DF – fundamental course, DD – field course, DS – specialty course, DC – complementary course, DAP – advanced study course, DSI – synthesis course, DCA – advanced knowledge course.*

*\*\* DI – mandatory course; DO – optional course.*

**3. Estimated workload (hours per semester)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3.1 Number of teaching hours/week | | 3 | of which:  3.2 course | 1 | 3.3 applications*\*\*\** | 2 |
| 3.4 Total of teaching hours within the program/semester | | 42 | of which:  3.5 lecture | 14 | 3.6 seminar | 28 |
| **3.7 Student workload for individual study** | | | | | | 108 |
| ***Distribution of workload*** | | | | | | [hours] |
| Individual study of texbooks, handbooks/reader, bibliography and notes | | | | | | 48 |
| Additional research (library, electronic resources, fieldwork) | | | | | | 32 |
| Homework (preparing seminar presentations, portfolios, critical essays, research papers, etc.) | | | | | | 20 |
| Individual consultations (optional) | | | | | | 0 |
| Evaluations / exams | | | | | | 8 |
| Other activities | | | | | | 0 |
| **3.8 Total hours per semester** | *42+108 = 150* | |  |  |  |  |
| **3.9 Number of credits** | 6 | |  |  |  |  |

*\*\*\* S - seminar; L - laboratory; P - project*

**4. Prerequisites (if any)**

|  |  |
| --- | --- |
| 4.1 Curriculum-related | Bachelors - Object Oriented Programming (optional) |
| 4.2 Skills-related | Programming skills (required), object-oriented programming (optional) |

**5. Requirements (if any)**

|  |  |
| --- | --- |
| 5.1. For running the course | Classroom available |
| 5.2. For running the seminar / laboratory /project  *\*The type is to be chosen according to the discipline* | Classroom/laboratory room available with computers and whiteboard |

**6. Acquired specific skills**

|  |  |
| --- | --- |
| Professional skills | Use of computer tools in an interdisciplinary context.  Applying the theoretical bases of informatics and formal models.  Development and implementation of computer applications using a modern object-oriented programming language (C ++ / C # / Java / J #). |
| Cross-cutting skills | Execution of complex professional tasks, in conditions of autonomy and professional independence, involving the detection and solution of related problems in the development of algorithms and data structures. Efficient use of information sources and communication resources as well as teamwork, in case of developing complex models. |

**7. Course goal and objectives**

|  |  |
| --- | --- |
| 7.1 The general objective of the course | The student will be familiar with the paradigms of object-oriented programming. |
| 7.2 Specific objectives | Creating and developing skills needed to develop complex, scalable and robust applications, based on the object-oriented programming paradigm using the Microsoft Visual Studio development platform and one of the C ++ / C # / Java / J # languages. |

**8. Contents**

|  |  |  |
| --- | --- | --- |
| **8.1 Lecture** | **Teaching methods** | **Number of hours** |
| **1. Introduction to object-oriented programming**  Object-oriented programming: definitions, classifications, particularities. | Interactive learning teaching methods  Dialogue  Problematization  Conversation  Methods that contribute to the development of critical thinking  Programs  Independent and cooperative learning | 2 hours |
| **2. Object-oriented paradigm**  Objects, Classes (Constructors, Access Modifiers, Methods), Class Hierarchies, Abstract Classes, Inheritance, Polymorphism, Interfaces. | 8 hours |
| **3. Handling errors by exceptions**  What is an exception ?, Detection and "catching" an exception, "Try" block, Standard exceptions, Building your own exceptions. | 2 hours |
| **4. I / O system**  Byte-level inputs and outputs, working with files, data flows, lexical analysis, object serialization. | 2 hours |
| **Bibliography:**   1. Bruce Eckel, *Thinking in Java*, 3rd ed. Revision 4.0. 2. Jawahar Puvvala, Alok Pota, *.NET for Java Developers: Migrating to C#*, Addison Wesley, 2003, ISBN 0-672-32402-4. 3. Irina Athanasiu, et. al., *Limbajul Java – O perspectiva pragmatica*, Ed. a II-a, Ed. Agora, 2000. 4. Galan C., Curs de C#, Ed. L&S Info-mat, 2008. 5. Mayo J., Microsoft Visual Studio 2010, A Beginer’s Guide, Ed. McGrow Hill, 2010. 6. Dan C., C# Object-oriented programmings, Ed. Apress, 2011. 7. Popovici D.M., Popovici I.M., C++. Tehnologia orientate spre obiecte. Aplicatii, Ed. Teora, Bucuresti, 2000 | | |

|  |  |  |
| --- | --- | --- |
| **8.2 Applications\* (seminar / laboratory / project)**  *\*The type is to be chosen according to the discipline* | **Teaching methods** | **Number of hours** |
| **1. Introduction to object-oriented programming**  Object-oriented programming: definitions, classifications, particularities. | Interactive learning teaching methods  Dialogue  Problematization  Conversation  Methods that contribute to the development of critical thinking  Programs  Independent and cooperative learning | 4 hours |
| **2. Object-oriented paradigm**  Objects, Classes (Constructors, Access Modifiers, Methods), Class Hierarchies, Abstract Classes, Inheritance, Polymorphism, Interfaces. | 16 hours |
| **3. Handling errors by exceptions**  What is an exception ?, Detection and "catching" an exception, "Try" block, Standard exceptions, Building your own exceptions. | 4 hours |
| **4. I/O system**  Byte-level inputs and outputs, working with files, data flows, lexical analysis, object serialization. | 4 hours |
| **Bibliography:**   1. Bruce Eckel, *Thinking in Java*, 3rd ed. Revision 4.0. 2. Jawahar Puvvala, Alok Pota, *.NET for Java Developers: Migrating to C#*, Addison Wesley, 2003, ISBN 0-672-32402-4. 3. Irina Athanasiu, et. al., *Limbajul Java – O perspectiva pragmatica*, Ed. a II-a, Ed. Agora, 2000. 4. Galan C., Curs de C#, Ed. L&S Info-mat, 2008. 5. Mayo J., Microsoft Visual Studio 2010, A Beginer’s Guide, Ed. McGrow Hill, 2010. 6. Dan C., C# Object-oriented programmings, Ed. Apress, 2011. 7. Popovici D.M., Popovici I.M., C++. Tehnologia orientate spre obiecte. Aplicatii, Ed. Teora, Bucuresti, 2000 | | |

**9. Correlation between the content of the course and the needs/expectations of the epistemic community, professional associations and/or significant employers relevant for the program**

|  |
| --- |
| Preparing graduates for direct and immediate insertion on the labor market in the IT or related field. |

**10. Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| Type of activity | 10.1 Evaluation criteria | 10.2 Evaluation methods | 10.3 Percentage of final grade |
| 10.4 Course | Acquiring the knowledge taught and skills to use the theoretical results in applications | Presence and active participation in the lecture | 10% |
| 10.5 Applications\*  (Seminar/Laboratory / Project)  *\*The type is to be chosen according to the discipline* | Activity in the laboratory | Presence and active participation in the laboratory | 50% |
| 10.6 Final examination | Exam | Student answer at examination subjects using a computer in a specific time interval | 40% |
| 10.7 Minimum standard of achievement for the acquisition of the ECTS credits: grade 5/10 | | | |
| Implementing a functional and correct application in a language oriented on a freely chosen object, according to the requirements expressed by the teacher in the exam ticket and read, understood and accepted by the student by participating in the exam. | | | |

Date of completion Course Instructor, Teaching Assistant,

20.09.2022 Prof.univ.dr. POPOVICI Dorin-Mircea Prof.univ.dr. POPOVICI Dorin-Mircea

Date of approval in the Department Head of Department,

27.09.2022 Conf. univ. dr. PUCHIANU Crenguța-Mădălina

Dean,

Conf. univ. dr. NICOLA Aurelian