# COURSE DESCRIPTION

1. **Course title:** GRAPHICAL PROGRAMMING

2. **Course code:**

3. **Validity of course description:** 2016/2017

4. **Level of studies:** MSc programme

5. **Mode of studies:** intramural studies

6. **Field of study:** MACRO COURSE (FACULTY SYMBOL)

7. **Profile of studies:** general

8. **Programme:** AUTOMATIC CONTROL

9. **Semester:** 2

10. **Faculty teaching the course:** Institute of Automatic Control, Rau1

11. **Course instructor:** Witold Nocoń, P.hD., D.Sc.

12. **Course classification:** programme courses

13. **Course status:** elective

14. **Language of instruction:** English

15. **Pre-requisite qualifications:** Good understanding of computer programming in any text-based programming language. It is assumed, that prior to commencing this course, students are able to implement complicated algorithms in any text-based programming language using all control statements (loops and conditionals), subprograms, arrays, pointers etc. It is also assumed that students are familiar with object-oriented programming concepts like encapsulation, inheritance, polymorphism.

16. **Course objectives:** The goal of this course is to gain practical knowledge of and ability to design, implement and test advanced and complex software systems for automation, control, monitoring, simulation, and modeling. Emphasis is put on using and modifying existing design patterns for designing scalable, reusable, easy to interpret and easy to debug software systems. The programming environment used in this course is LabVIEW.

17. **Description of learning outcomes:**

<table>
<thead>
<tr>
<th>Nr</th>
<th>Learning outcomes description</th>
<th>Method of assessment</th>
<th>Teaching methods</th>
<th>Learning outcomes reference code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The student knows basic design patterns (state, machine, queued state machine) and synchronization mechanisms (queues, events etc.)</td>
<td>CL</td>
<td>Lecture/Laboratory</td>
<td>K_W03</td>
</tr>
<tr>
<td>2.</td>
<td>The student knows the multithreaded design patterns (master/slave, producer/consumer) and object-oriented design patterns</td>
<td>CL</td>
<td>Lecture/Laboratory</td>
<td>K_W15</td>
</tr>
<tr>
<td>3.</td>
<td>The student can implement basic design patterns in PC and PAC environments.</td>
<td>CL, PS</td>
<td>Laboratory</td>
<td>K_U09</td>
</tr>
<tr>
<td>4.</td>
<td>The student can implement a class hierarchy for a given problem and use objects of those classes in an application.</td>
<td>CL, PS</td>
<td>Laboratory</td>
<td>K_U09</td>
</tr>
<tr>
<td>5.</td>
<td>The student is capable of selecting optimal design patterns for given problems and implement those in PC and PAC environments (including real-time environments) and SCADA systems</td>
<td>CL, PS</td>
<td>Laboratory</td>
<td>K_W07 K_W19 K_U23</td>
</tr>
<tr>
<td>6.</td>
<td>The student is capable of stating the benefits of using object-oriented solutions to programming for the given problems in PC and PAC environments.</td>
<td>PS, OS</td>
<td>Lecture</td>
<td>K_U13</td>
</tr>
</tbody>
</table>

18. **Teaching modes and hours**

*Lecture / BA / MA Seminar / Class / Project / Laboratory*

Sem 2 - 30 h., Sem 15 - 30 h.
19. Syllabus description:

Semester 5:

Lecture:


4. SubVIs. Creating and using SubVIs. VI properties. Reentrant execution. Dynamically loaded VIs.

5. Local and global variables. Creating, using global/local variables. Mechanical action of buttons vs. local/global variables.


7. Event based programming. Using the event structure. Signalizing and filter events. Practical rules of using the event structure. Methods for programmatic firing of events (two methods)


13. Networking and communication protocols. Shared variables, TCP, UDP, POP, SMTP etc.

14. Programmable Automation Controllers, Real-Time Targets (compactRIO for example).

15. SCADA systems.

Laboratory:

1. Fundamentals 1. Using basic programming structures, controlling execution of the program. using numerical, Boolean, string values. Programming simple mathematical computations, iterative computations etc.


3. Advanced task 1. Writing a data acquisition application using a DAQ device (for example a NI-USB-6008). Displaying data to the user in a configurable way. Data archive on disk.

4. Advanced task 2. Writing a Communications application using TCP/IP. Sending and receiving data over the network. Designing simple communication protocols.

5. Advanced task 3. Writing an application for control and monitoring of an industrial plant. Using for example ComactRIO controllers to read measurement data, write, control values, implement closed loop control systems, automatic and manual control, programming alarming functions etc.

20. Examination: No

21. Primary sources:


22. Secondary sources:

23. Total workload required to achieve learning outcomes

<table>
<thead>
<tr>
<th>Lp.</th>
<th>Teaching mode</th>
<th>Contact hours / Student workload hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lecture</td>
<td>30/10</td>
</tr>
<tr>
<td>2</td>
<td>Classes</td>
<td>/</td>
</tr>
<tr>
<td>3</td>
<td>Laboratory</td>
<td>15/35</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>/</td>
</tr>
<tr>
<td>5</td>
<td>BA/MA Seminar</td>
<td>/</td>
</tr>
<tr>
<td>6</td>
<td>Other</td>
<td>5/5</td>
</tr>
<tr>
<td></td>
<td>Total number of hours</td>
<td>50/50</td>
</tr>
</tbody>
</table>

24. Total hours: 100

25. Number of ECTS credits: 4

26. Number of ECTS credits allocated for contact hours: 2

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 2

26. Comments:

Approved:

……………………………………………………………
(date, Instructor's signature)

……………………………………………………………
(date, the Director of the Faculty Unit signature)