AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Module name: Object oriented programming

Academic year: 2013/2014 Code: RMS-1-202-s ECTS credits: 3

Faculty of: Mechanical Engineering and Robotics

Field of study: Mechatronics with English as instruction languagege Specialty: —

Study level: First-cycle studies Form and type of study: Full-time studies

Lecture language: English Profile of education: Academic (A) Semester: 2

Course homepage: —

Responsible teacher: dr inż. Miękina Lucjan (miekina@agh.edu.pl)

Academic teachers: dr Pluta Marek (pluta@agh.edu.pl)
dr inż. Miękina Lucjan (miekina@agh.edu.pl)

Description of learning outcomes for module

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Connections with FLO	Method of learning outcomes verification (form of completion)
Social competence			
M_K001	can study compound matters of informatics, regarding their critical importance with respect to practical applications	MS1A_K04	Execution of laboratory classes
M_K002	can analyze the requirements for a programming task nad properly organize the steps necessary to complete the project	MS1A_K05	Execution of laboratory classes
Skills			
M_U001	can apply the principles of object-oriented approach to programming applications	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U002	can write, build and run a C++ program based on object-oriented paradigm - that is, one which defines classes and uses objects	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U003	can define a primary class (possibly abstract) and derived class which implements polymorphism	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U004	can write basic operator functions, e.g. +, =, ==, !=, >=, <=, [], implemented as class members or as external friend functions; can use them in a client code	MS1A_U14	Test, Execution of laboratory classes, Test results

M_U005	can apply the template mechanism to define universal containers (array, vector) capable of holding data of any type	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U006	can define stream operators << and >>, operating on objects of any class and streams of any type	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U007	can define and use static class members, friend functions and classes, nested classes	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U008	can handle system and software exceptions, as well as uniformly handle exceptions of both kinds while programming in C++	MS1A_U14	Test, Execution of laboratory classes, Test results
M_U009	can handle dynamic data structures (vectors, lists, queues, maps) using STL in C++ programs	MS1A_U14	Test, Execution of laboratory classes, Test results
Knowledge			
M_W001	knows and understands the principles of object-oriented programming, as well as the advantages it brings	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W002	knows and understands the general structure of a C++ program	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W003	knows and understands the notions of object and class (its purpose, members and their visibility, C++ declaration syntax)	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W004	knows and understands the notions of abstract class and derived class, their place in the software development process, their purpose and way of defining (including virtual metods)	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W005	knows and understands the operator functions, their purpose, ways of defining and advantages of using them	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W006	knows and understands the mechanism of templates (of functions and classes), its purpose and syntax	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W007	knows and understands the stream input and output operators (<< and >>), including the way of defining them for a given class and stream type	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W008	knows and understands static class members, friend functions and classes, nested classes	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W009	knows and understands the C++ exception handling mechanism, its purpose and importance for software reliability; knowns and understands the way of handling system and software exceptions, as well as the unified approach to handling exceptions of both kinds	MS1A_W10	Test, Execution of laboratory classes, Test results
M_W010	knows the standard C++ library and understands how to handle dynamic data structures (vectors, lists, queues, maps) using STL	MS1A_W10	Test, Execution of laboratory classes, Test results

FLO matrix in relation to forms of classes

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Form of classes

		Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Others	Fieldwork classes	Workshops	E-learning
Social competence												
M_K001	can study compound matters of informatics, regarding their critical importance with respect to practical applications	-	-	-	-	-	-	-	-	-	-	-
M_K002	can analyze the requirements for a programming task nad properly organize the steps necessary to complete the project	-	-	-	-	-	-	-	-	-	-	-
Skills												
M_U001	can apply the principles of object-oriented approach to programming applications	-	-	+	-	-	-	-	-	-	-	-
M_U002	can write, build and run a C++ program based on object-oriented paradigm - that is, one which defines classes and uses objects	-	-	+	-	-	-	-	-	-	-	-
M_U003	can define a primary class (possibly abstract) and derived class which implements polymorphism	-	-	+	-	-	-	-	-	-	-	-
M_U004	can write basic operator functions, e.g. +, =, ==, !=, >=, <=, [], implemented as class members or as external friend functions; can use them in a client code	-	-	+	-	-	-	-	-	-	-	-
M_U005	can apply the template mechanism to define universal containers (array, vector) capable of holding data of any type	-	-	+	-	-	-	-	-	-	-	-
M_U006	can define stream operators << and >>, operating on objects of any class and streams of any type	-	-	+	-	-	-	-	-	-	-	-
M_U007	can define and use static class members, friend functions and classes, nested classes	-	-	+	-	-	-	-	-	-	-	-
M_U008	can handle system and software exceptions, as well as uniformly handle exceptions of both kinds while programming in C++	-	-	+	-	-	-	-	-	-	-	-

M_U009	can handle dynamic data structures (vectors, lists, queues, maps) using STL in C++ programs	-	-	+	-	-	-	-	-	-	-	-
Knowledge												
M_W001	knows and understands the principles of object-oriented programming, as well as the advantages it brings	+	-	-	-	-	-	-	-	-	-	-
M_W002	knows and understands the general structure of a C++ program	+	-	-	-	-	-	-	-	-	-	-
M_W003	knows and understands the notions of object and class (its purpose, members and their visibility, C++ declaration syntax)	+	-	-	-	-	-	-	-	-	-	-
M_W004	knows and understands the notions of abstract class and derived class, their place in the software development process, their purpose and way of defining (including virtual metods)	+	-	-	-	-	-	-	-	-	-	-
M_W005	knows and understands the operator functions, their purpose, ways of defining and advantages of using them	+	-	-	-	-	-	-	-	-	-	-
M_W006	knows and understands the mechanism of templates (of functions and classes), its purpose and syntax	+	-	-	-	-	-	-	-	-	-	-
M_W007	knows and understands the stream input and output operators (<< and >>), including the way of defining them for a given class and stream type	+	-	-	-	-	-	-	-	-	-	-
M_W008	knows and understands static class members, friend functions and classes, nested classes	+	-	-	-	-	-	-	-	-	-	-
M_W009	knows and understands the C++ exception handling mechanism, its purpose and importance for software reliability; knowns and understands the way of handling system and software exceptions, as well as the unified approach to handling exceptions of both kinds	+	-	-	-	-	-	-	-	-	-	-
M_W010	knows the standard C++ library and understands how to handle dynamic data structures (vectors, lists, queues, maps) using STL	+	-	-	-	-	-	-	-	-	-	-

Module content

Lectures

The lecture covers:
- principles of object-oriented programming, and the advantages it brings
- general structure of a C++ program
- notions of object and class (its purpose, members and their visibility, C++ declaration syntax)
- notions of abstract class and derived class, their place in the software development process, their purpose and way of defining (including virtual metods)
- operator functions, their purpose, ways of defining and advantages of using
- mechanism of templates (of functions and classes), its purpose and syntax
- stream input and output operators (<< and >>), including the way of defining them for a given class and stream type
- static class members, friend functions and classes, nested classes
- C++ exception handling mechanism, its purpose and importance for software reliability; handling system and software exceptions, as well as the unified approach to handling exceptions of both kinds
- standard C++ library and its application to dynamic data structures (vectors, lists, queues, maps)

Laboratory classes

During laboratories the students first discuss the ideas presented in the lectures, then apply these ideas to solve the scheduled problems. The solution consists of a set of classes written in order to support the needs of a program (e.g. to satisfy its requirements)

Method of calculating the final grade

Grades are calculated as follows:
- software labs 50 %
- quizz 25 %
- final test 25 %

All of them must be passed.

Prerequisites and additional requirements

Prerequisites and additional requirements not specified

Recommended literature and teaching resources

- B. Stroustrup: The C++ Programming Language, Addison-Wesley, 2000
- B. Stroustrup: Programming — Principles and Practice Using C++, Addison-Wesley, ISBN 978-0321543721. December 2008.
- B. Eckel: Thinking in C++, Prentice Hall, 2002
- Stroustrup’s web page: http://www.research.att.com/~bs/

Scientific publications of module course instructors related to the topic of the module

Additional scientific publications not specified

Additional information

None

Student workload (ECTS credits balance)

Student activity form	Student workload
Participation in lectures	15 h
Preparation for classes	28 h
Participation in laboratory classes	28 h
Realization of independently performed tasks	15 h
Examination or Final test	2 h
Summary student workload	88 h
Module ECTS credits	3 ECTS