THE UNIVERSITY OF BUCKINGHAM

Sarajevo School of Science and Technology

MODULE SPECIFICATION

Name of Module		Programming and Problem Solving II					
Parent School/Dept		Computer Science/Information Systems					

Programme(s) where module is offered		BSc Computer Science with Electrical Engineering;
		BSc Computer Science with Economics;
		BSc Computer Science with Business;
		BSc Computer Science with International Relations;
		BSc Computer Science with Political Science;
		BSc Information Systems with Electrical Engineering;
		BSc Information Systems with Economics;
		BSc Information Systems with Business;
		BSc Information Systems with International Relations;
		BSc Information Systems with Political Science;					

Status (core, option, free choice)		Core		**Pre-Requisite Modules or Qualifications**		none	
FHEQ Level	4	**Unit Value**	8 ECTS	**Module Code**	**CS160**	**Module coordinator**	Dr. Željko Jurić
Term taught		Spring		**Applicable From**		2016	

Educational Aims of the Module

The main aim of the module is to introduce the more advanced style of programming, including application of modern programming style for solving problems using a computer. Lectures build upon this basis and provide intermediate to advanced programming knowledge in C++ programming language. Laboratory work and programming assignments are an integral part of this module.

Module Outline/Syllabus

- Exceptions and exception handling.
- Pointers and pointer arithmetic.
- Dynamic memory allocation and memory management.
- Structures, classes and objects. Attributes and methods. Static attributes. Encapsulation and information hiding.
- Constructors and destructors. Shallow and deep copies. Copy constructors. Assignment operator.
- Operator overloading.
- Inheritance. Polymorphism. Virtual functions. Abstract base classes.
- Stream-based file handling.
- Templates. Generic functions and classes.
- Standard library containers and iterators (vectors, deques, lists, sets, etc.)

Student Engagement Hours

Type	Number per Term	Duration	Total Time
Lectures	30	2 hours	60 hours
Laboratory sessions	15	2 hours	30 hours
		Total Guided/Independent Learning Hours	**110 hours**
		Total Contact Hours	**90 hours**
		Total Engagement Hours	**200 hours**

Assessment Method Summary

Type	Number Required	Duration / Length	Weighting	Timing/Submission Deadline
Final exam	1	180 minutes	50%	End of semester
Mid-term exam	1	90 minutes	20%	Mid-semester (8th week)
Project (individual)	1	2,000 words	10%	14th week
Test	2	60 minutes	20%	5th and 13th week

Module Outcomes		
Intended Learning Outcomes: 1. Program effectively using advanced C++ concepts, including Object Oriented and Generic style 2. Theoretical knowledge of different programming paradigms 3. Make knowledgeable programming and policy decisions	→	**Teaching and Learning Strategy:** (ILO:1-3) 1. Lectures and laboratory sessions are going to be delivered containing the material from the module outline 2. Regular presentation of solutions with peer feedback and discussion are encouraged both during lecture time and especially during lab time 3. Lectures and practical session assignments
	→	**Assessment Strategy** 1. Mid-term exam (ILO:1) 2. Final exam (ILO:1-3) 3. Test (ILO:1, 3) 4. Project (ILO:1-3)
Practical Skills 1. Practical programming in C++ 2. Ability to design algorithms for solving basic problems 3. Ability to design basic projects in an object-oriented language in a teamwork environment 4. Ability to organize a good technical presentation	→	**Teaching and Learning Strategy:** 1. Lab exercises with tutor-lead support (PS:1-4) 2. Individual project assignment (PS:1-4) 3. Use of subject tests (PS:1-4)
	→	**Assessment Strategy** 1. Mid-term exam (PS:1) 2. Final exam on Computer (PS:1-2) 3. Project (PS:1-4) 4. Test (PS:1-2)
Transferable Skills 1. Ability to discuss, accurately, basic design issues 2. Ability to intelligently present technical solutions in both written and verbal formats 3. Presentation skills 4. IT skills	→	**Teaching and Learning Strategy:** 1. Lab exercises with tutor-lead support (TS:1,4) 2. Individual project assignment (TS:1-4)
	→	**Assessment Strategy** 1. Final Exam on Computer (TS:2, 4) 2. Test (TS:2,4) 3. Project (TS:1-4)

Key Texts and/or other learning materials

Set text:
Liang, D., (2013), "Introduction to Programming with C++", 3rd edition, Pearson Education Ltd

Supplementary Resources:

- Stroustrup, B., (2014), Programming: Principles and Practice using C++, 2nd Edition, Addison Wesley
- Alessandrini, V., (2015), Shared Memory Application Programming, Morgan Kauffman
- Elsevier, (2015), Science of Computer Programming, Open Archive [online], http://www.journals.elsevier.com/science-of-computer-programming/open-archive/ (Accessed 25th November 2015).

Please note: This specification provides a concise summary of the main features of the module and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if he/she takes full advantage of the learning opportunities that are provided. More detailed information on the learning outcomes, content and teaching, learning and assessment methods of each module and programme can be found in the departmental or programme handbook. The accuracy of the information contained in this document is reviewed annually by the University of Buckingham and may be checked by the Quality Assurance Agency.

Date of Production	Autumn 2016
Date approved by School Learning and Teaching Committee	28th September 2016
Date approved by School Board of Study	12th October 2016
Date approved by University Learning and Teaching Committee	2nd November 2016
Date of Annual Review	December 2017