

| | |
|---|---|
| SUBJECT: | Programming Language |
| ECTS CREDITS: | 4 ECTS |
| TEACHERS, UNIVERSITY AND EMAILS: | <ol style="list-style-type: none"> 1. Razzakov Shavkat, (BETI, raz_shi@rambler.ru) 2. Ibragimov Ulugbek (BETI, ulugbek.ibr@gmail.com, ciulugbek@list.ru) 3. Fayziev Shavkat (BETI, shavkatfayz@rambler.ru) |
| RESPONSIBLE TEACHER: | Razzakov Shavkat |
| LANGUAGE OF INSTRUCTION: | Russian (Ibragimov Ulugbek) English (Fayziev Shavkat) |
| ACADEMIC COURSE: | 2016—17. |
| NAME OF THE MASTER'S DEGREE: | Master's in Mathematical Engineering. |

COURSE AIMS:

| |
|---|
| <ul style="list-style-type: none"> • fundamental computational concepts underlying most programming languages; • a range of problem solving techniques using computers; • the role of programming within the overall software development process; • attitudes and working practices appropriate for a professional programmer; • To be competent using the most common features in MATLAB; • To understand how to use MATLAB to solve engineering problems; • to provide an introduction to programming in MATLAB for those with some programming experience so that the attendee is able to write MATLAB programs and will have the ability to find more advanced MATLAB tools for themselves. |
|---|

LEARNING OUTCOMES:

| |
|--|
| <p>Knowledge - know about:</p> <ul style="list-style-type: none"> • techniques for solving problems; • basic computational concepts and elementary data structures; • the edit-compile-link-run cycle from a user point of view; • testing strategies; • the main activities of software development and their interactions, and |
|--|

some of the major problems of software development.

Skills - be able to:

- hand-execute simple programs, showing how input data is processed, output data is produced, and how the values of internal variables change;
- explain at various levels the behaviour of fragments of programming language code;
- amend existing programs to adjust or correct their functionality;
- translate well-structured plans into working programs;
- analyse simple problems involving text, numbers and graphics;
- use the error messages of the compiler to identify and correct mistakes in program syntax;
- use testing strategies to identify and correct semantic errors in programs;

Attitudes - appreciate that:

- a programmer requires creativity in order to solve problems and precision in the construction and manipulation of programming language code;
- a programmer builds up a repertoire of techniques for solving problems, usually adapting and reusing techniques as each new problem is encountered;
- a programmer must be able to communicate his/her ideas to others;
- effective programming requires effort both in front of and away from a computer;
- learning to program requires commitment and perseverance;
- provide programming skills using an object-oriented technology in the programming language C ++;
- In addition to studying the language itself, the program includes consideration of various algorithms, often used in programming;
- The course is designed for training, retraining and additional training of personnel in Mathematical - engineering and IT engineering;

Programming Matlab course

- Write simple program modules to implement single numerical methods and algorithms;
- Calculate solutions to mathematical engineering problems using

standard numerical methods;

- Test program output for accuracy using hand calculations and debugging techniques;
- Analyze the applicability and accuracy of numerical solutions to diverse mathematical engineering problems;
- Synthesize multiple program modules into larger program packages;
- Distill numerical results into a readable format that answers specific mathematical engineering analysis and design questions.

COURSE SYLLABUS:

The C++ programming language

- Introduction to programming in C++;
- Basic data types.
Keyboard and files I/O Control sentences. Dynamic memory management: pointers. Structures. Functions. Overloading.

Object Oriented Programming in C++.

- Introduction to Object Oriented Programming (OOP);
- Classes and instances (objects);
- Operator overloading;
- Friend functions and classes;
- Inheritance;
- Polymorphism;
- Templates.

Standard Template Library (STL).

- Introduction to STL;
- STL containers and iterators;
- Basic STL containers management;

Introduction to Programming Matlab

- Introduction to Matlab. Basic commands and functions;
- Vectors and Matrices in Matlab. Sparse matrices. Plots;
- Files. M and programming. Data Structures in Matlab;
- Introduction to Fortran 90: Data types and control flow;
- Arrays in Fortran 90. Procedures, modules and interfaces;
- Input / Output data in Fortran 90.

LITERATURE:

- Basic literature:
 - [1] C++ for Engineers and Scientists. Gary J. Bronson. 2015;
 - [2] Beginning Programming with C For Dummies. Dan Gookin. 2014;
 - [3] Starting Out with C++: Early Objects. Tony Gaddis, Judy Walters, Godfrey Muganda. 2013;
 - [4] Advanced C and C++ Compiling. Milan Stevanovic. 2014;
 - [5] Professional C++. Marc Gregoire. 2014;
 - [6] The C++ Programming Language. Bjarne Stroustrup. 2013;
 - [7] Data Structures and Algorithms in C++. Michael T. Goodrich, Roberto Tamassia, David M. Mount. 2011;
 - [8] STL Tutorial and Reference Guide: C++ Programming. Дэвид Р. Мюссер, Жилмер Дж. Дердж. 2010;
 - [9] Николай Джосьютис. C++ Стандартная библиотека. Для профессионалов. 2004;
 - [10] Эффективное использование STL. Скотт Мейерс, 2002;
 - [11] Harvey M. Deitel, Paul J. Deitel (2009). C++ : cómo programar (6^a ed.). Pearson Educación;
 - [12] Walter Savitch. Problem Solving with C++: The Object of Programming, Fifth Edition . Addison-Wesley, 2004;
 - [13] Ray Lischner. C++ In a Nutshell. O'Reilly Media, 2003;
 - [14] M. Metcalf, J.K. Reid. Modern Fortran Explained Oxford University Press, 2011;
 - [15] S.J. Chapman, Fortran 90/95 for scientists and engineers. WCB/McGrawHill, 2004;
 - [16] Лазарев Ю. Моделирование процессов и систем в Matlab. Учебный курс, 2005 г.;
 - [17] Holly More. MATLAB for engineers. Source. 2012
 - [18] Are Mjaavatten Bernt Lie. Introduction to Matlab for Engineers. 2005;
- Further reading:
 - [1] Самоучитель C++. Шилдт Г. 2003 г.;
 - [2] Освой самостоятельно C++ за 24 часа. Либерти Д., Хорват Д. 2007 г.;
 - [3] Язык программирования C++. Страуструп Б. 2002 г. ;
 - [4] Язык программирования C++: Специальное издание. Страуструп Б. 2008 г.;
 - [5] Библиотека программиста C++. Эджер Дж. 2000 г.;
 - [6] Объектно-ориентированное программирование в C++. Лафоре Р. 2004 г.;
 - [7] Фундаментальные алгоритмы на C++ ч.1-4.. Седжвик Р. 2000 г.;
 - [8] Фундаментальные алгоритмы на C++ ч.5: Алгоритмы на графах. Седжвик Р. 2000 г.;
 - [9] В. П. Дьяконов. MATLAB 7.*/R2006/R2007. Самоучитель. 2008 год. 768 стр. PDF. 8.7 Мб.
 - [10] Дашенко и др. MATLAB в инженерных и научных расчетах. 2003 год. 215 стр.

- [11] Чен К, Джиблин П., Ирвинг А. Matlab в математических исследованиях. 2001 год.
- [12] "MATLAB. Самоучитель. Практический подход". А. Васильев. 2012;

TEACHING METHOD:

Face-to-face lectures, tutorial and lab classes, online lecture notes, tutorial solutions, peer mentoring, 'help day' by tutors/lecturers and selection of textbooks having interactive solutions of numerical problems. Videoconference to all CA partners.

METHOD OF ASSESSMENT:

70% for **Knowledge and understanding:**

- written examination which includes examination on theory taught in the lectures and practical/seminar/laboratory classes

30% for **Practical issues:**

- includes solved exercises on personal studies,
- attendance to the lab/practical/seminar classes
- Oral and written responses based on individual experience
- Creating programs using a programming language associated mathematical engineering
- The use of programming language in real life situations.

Method of REASSESSMENT:

70% for **Knowledge and understanding:**

- written examination which includes examination on theory taught in the lectures and practical/seminar/laboratory classes

30% for **Practical issues:**

- written examination which includes solved exercises during the reassessment examination

STUDENT WORKLOAD:

On-site work at the classroom (attendance to classes and participation on them) = 80 hours.

Lecture hours: 2 hours once every two weeks

Practical/Lab hours: 2 hours once every two weeks

Mid term exam: 2 academic hours

Final exam: 3 academic hours

Self-study (autonomous study, doing exercises, programming, recommended readings)

= 64 hours.

RECOMMENDATIONS:

<http://www.learnerstv.com/Free-Computer-Science-video-lecture-courses.htm>
Welcome to Learnerstv.com. This is a comprehensive site providing thousands of downloadable Video lectures, Live Online Tests, etc in the fields of Mathematics, Computer Science, Engineering and etc FREE to its visitors... This site provides free video and audio lectures of whole courses conducted by faculty from reputed universities around the world. Science Animations provide students with fun and innovative ways of learning. Free live timed online tests with instant feedback and explanations will help you refine your test taking skills. Most of the materials offered are licensed by the respective institutes under a Creative Commons License.

<http://freevidelectures.com/Subject/Programming>
<http://freevidelectures.com/Course/3321/Introduction-to-MATLAB>
Start here. Learn any programming language. Includes everything from C programming, C++, C#, JAVA, AJAX, .NET(dot net) to scripting languages PERL, Python, COBOL. Courses cover introductory programming to advanced level.
Introduction to MATLAB.

OTHER COMMENTS: