**Government of Russian Federation**

**Federal State Autonomous Educational Institution of High Professional Education**

**“National Research University Higher School of Economics”**

**Syllabus for the course**

**“A Theorist’s Toolkit”**

bachelor program in 01.03.02.62 “Applied Mathematics and Information Science”

Authors:

Vladimir Podolskii, assistant professor, vpodolskii@hse.ru

Moscow - 2018

*This program cannot be used by other departments and other universities without the author’s permission.*

# Course description

a) Title: a theorist’s toolkit.

b) Pre-requisites:
- Discrete mathematics, linear algebra, probability theory, theory of computing, computational complexity theory.
- Basic English language, both oral and written.

c) Course type: for 4th year Bach.

d) Abstract:

Once some area arrives at the problems that are not clear how to solve, it becomes especially important not only to obtain new results, but also to develop new methods and techniques.
In recent decades Theoretical Computer Science had expanded substantially and the number of mathematical tools used in it in it rapidly increased. The goal of this course is to go through these tools and techniques, so that students can apply them when needed.

This program establishes the minimal requirements to students’ knowledge and skills and determines the content of the course and educational techniques used in teaching the course. The present syllabus is aimed at faculty teaching the course and undergraduate students studying 01.03.02.62 “Applied Mathematics and Information Science”. This syllabus meets the standards required by:

* Educational standards of National Research University Higher School of Economics;
* Bachelor educational program for 01.03.02.62 “Applied Mathematics and Information Science”.

# Learning Objectives

After this course, students will understand the following concepts:

* Basics of Fourier analysis over finite fields and its applications;
* Probabilistic methods in Computer Science, concentration of the measure, local lemma;
* Polynomial method;
* Linear-algebraic methods.

# Main Competencies Developed after Completing the Study of This Discipline

After completing the study of the discipline the student will:

• understand the concepts listed above,

• be able to critically analyse resources used by a program (and optimize them at a high level),

• be able to recognize intractable problems and categorize their difficulty,

• have deeper understanding and trained problem solving skills of known materials in: algebra, probability theory, discrete math, algorithms

• be trained to read, speak (and possibly write) technical and mathematical English.

After completing the study of the discipline the student should have developed the following competencies:

|  |  |  |  |
| --- | --- | --- | --- |
| **Competence**  | **Code**  | **Descriptors (indicators of achievement of the result)** | **Educative forms and methods aimed at generation and development of the competence** |
|  Capable to identify the scientific nature of problems in the professional field of activity | UC-2 | Students obtain necessary knowledge to understand and formulate the theoretical difficulty of problems they are solving.  | Lectures and exercise sessions |
| Capable to solve problems in the professional field of activity using analysis and synthesis | UC-3 | Students will be able to solve problems using methods of theoretical computer science  | Lectures and exercise sessions |
| Capable to describe problems and situation of the professional field of activity using mathematical language and methods | PC-1 | Students will be able to formulate and identify problems in Computational Complexity Theory | Lectures and exercise sessions |
|  Capable to comprehend, modify and apply contemporary mathematical methods | PC-3 | Students will be able to apply, combine and modify standard methods in theoretical computer science | Lectures and exercise sessions |
| Capable to build a mathematical model and perform its analysis for the specified theoretical or applied problem | PC-8 | Students will be able to apply standard models of theoretical computer science to problems in this area | Lectures and exercise sessions |

# Course plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Topic | Total hours | Contact hours | Self-study |
| Lectures | Seminars | Practice lessons |
| 1 | Fourier analysis over finite fields | 16 | 10 | 10 |  | 50 |
| 2 | Probabilistic methods | 44 | 4 | 4 |  | 24 |
| 3 | Polynomial method | 16 | 4 | 4 |  | 24 |
| 4 | Linear-algebraic methods | 30 | 2 | 2 |  | 14 |
|  | Total | 152 | 20 | 20 |  | 112 |

**Topic 1. Fourier analysis over finite fields.**

Basics of Fourier analysis over finite fields and its applications in various fields of Theoretical Computer Science

**Topic 2. Probabilistic methods.**

Probabilistic methods in Computer Science, concentration of the measure, local lemma.

**Topic 3. Polynomial method.**

Techniques based on polynomials in Boolean circuit complexity.

**Topic 4. Linear-algebraic methods.**

Applications of linear algebra in computational complexity theory and combinatorics.

# Reading list

*a. Required*

* Ryan O'Donnell: **Analysis of Boolean Functions.** Cambridge University Press 2014, ISBN 978-1-10-703832-5, pp. I-XX, 1-423. <http://www.contrib.andrew.cmu.edu/~ryanod/?page_id=2334>

*b. Recommended*

* 1. Noga Alon, Joel Spencer: **The Probabilistic Method.** John Wiley 1992, ISBN 0-471-53588-5
	2. Devdatt P. Dubhashi, Alessandro Panconesi: **Concentration of Measure for the Analysis of Randomized Algorithms.** Cambridge University Press 2009, ISBN 978-0-521-88427-3, pp. I-XIV, 1-196
	3. Stasys Jukna: **Extremal Combinatorics - With Applications in Computer Science.** Texts in Theoretical Computer Science. An EATCS Series, Springer 2011, ISBN 978-3-642-17363-9, pp. 1-411
	4. Stasys Jukna: **Boolean Function Complexity - Advances and Frontiers.** Algorithms and combinatorics 27, Springer 2012, ISBN 978-3-642-24507-7, pp. I-XV, 1-617
	5. Troy Lee, Adi Shraibman: **Lower Bounds in Communication Complexity.** Foundations and Trends in Theoretical Computer Science 3(4): 263-398 (2009)

#  Grading system

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Form | 1 year | Details  |  |  |
| 1 | 2 | 3 | 4 |  |
| Intermediate | Colloquium |  |  | 1 |  | Oral discussion of the theoretical material. Near the end of the third module. Grade - *Оcol .* |
| Homework |  |  | 1 |  | A homework broken into pieces with weekly deadlines. The grade is proportional to the number of solved problems. Grade *Оhw*. |
| Final | Exam  |  |  | 1 |  | Written exam, 2h40m, Grade – *Оexam*.  |

# The grade formula

*Оfinal =* 0,7*·Оaccum* + 0,3*·Оexam*,

where *Оaccum* = (1/2)*Оcol* +(1/2)*Оhw*.

***In other words:***

***Homeworks*** count for 35% of the final grade in total.

***The colloquim*** is worth 35% of the final grade.

***The final exam*** is worth 30% of the final grade.

All intermediate grades are computed without rounding. Rounding is applied only to the final grade. The following rounding is applied. Grades between 1 and 5 are rounded down. Grades between 0 and 1 as well as grades between 6 and 10 are rounded up. Grades between 5 and 6 are rounded arithmetically.

**Table of Grade Accordance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ten-point****Grading Scale** |

|  |  |
| --- | --- |
|  |  |

**Five-point****Grading Scale** |  |
| 1 - very bad2 – bad3 – no pass | Unsatisfactory - 2 | **FAIL** |
| 4 – pass5 – highly pass | Satisfactory – 3 | **PASS** |
| 6 – good7 – very good | Good – 4 |
| 8 – almost excellent9 – excellent10 – perfect | Excellent – 5 |

1. **Guidelines for Knowledge Assessment**

The final exam consists of a selection of problems. Students are allowed to use textbooks and notes. Each question will require to solve mathematical problems using materials presented during the lectures. Questions will be asked in English and students can answer either in English or in Russian.

To be prepared for the final exam, students will be given 5-10 exercises every week (during the seminar). They can ask for hints and feedback on solutions.

1. **Methods of Instruction**

There will be *theory lectures*, of 2 accademic hours each, during which conceptual ideas are explained.

Each lecture is followed by a *seminar* of 2 academic hours, in which students solve exercises that deepen the understanding of the materials and train problem solving skills. Each weak a list of 6-10 exercises are given.

Every week 2-4 exercises are given for homework to be submitted next week.

The student can consult the professors and during the office hours. Intensive guidance can be given during the office hours of the teachers.