1 Scope of Use

This program establishes the minimal requirements to students' knowledge and skills and defines the content of the discipline and the types of studies and reports.

The present syllabus is aimed at faculty staff teaching the course «Combinatorics, Graphs and Computational Logic», teaching assistants and students studying 01.03.02 "Applied Mathematics and Informatics" of the educational program "Applied Mathematics and Informatics".

This syllabus meets the standards required by:

- Educational standards of Federal State Autonomous Educational Institution of High Professional Education "National Research University "Higher School of Economics";
- Bachelor educational program "Applied Mathematics and Informatics" for 01.03.02 "Applied Mathematics and Informatics";
- University curriculum of the educational program for 01.03.02 "Applied Mathematics and Informatics" training direction of the "Applied Mathematics and Informatics" bachelor program, approved in 2018.

2 Course Description

a. Title of a Course

The discipline is called «Combinatorics, Graphs and Computational Logic». It consists of three parts:

- Bioelectrical digital signal processing
- Introduction to the Semantic Web Technologies
- Pre-defense of the course work

b. Pre-requisites

The following knowledge and competences are useful for better understanding of the course:

- basic C/C++ programming skills;
- basic notions from the first course of "Discrete mathematics".

No special skills are required, but all students are advised to have good fundamental mathematical level.

c. Course Type

This is a compulsory course for specialization "Data Analysis and Intelligent Systems", 01.03.02 "Applied Mathematics and Informatics".

d. Abstract

«Combinatorics, Graphs and Computational Logic» **class covers more complicated sections of discrete mathematics.**

The combinatorics chapter is devoted to in-depth combinatorics, recursive sequences and the group action on the sets. The graph chapter covers theoretical foundations of algebric graph theory, algorithms on graphs and their applications. Computational Logic chapter covers modern methods for closed classes of Boolean logic, propositional calculus, predicate logic, and properties of classes to have a system of identities, as well as Prolog and Resolution Method. Cryptography applications is not included for the purpose to not interfere with parallel courses. As a result, the CGCL greatly advances students' knowledge in the fields of modern discrete mathematics theory and applications, preparing our students to professional work in research projects and IT-industry. So, despite its theoretical content, the class has got many applications in the theory of algorithms, borrowing many math/IT concepts.

This course may be useful for collaboration with the following disciplines:

- Information theory
- Ordered sets and lattices
- Applied Graph Theory
- Computer Algebra
- Semantic Web

The importance of discrete mathematics has increased significantly since last 50 years, although there are few full handbooks of modern methods in its areas. The purpose of the course **CGCL** is to provide systematic review of certain fields of discrete mathematics for computer scientists, mathematicians, and others, such as students, physical and social scientists, who need information about discrete mathematics.

The scope of material includes the many areas generally considered to be parts of discrete mathematics, focusing on the information considered essential to its application in computer science and engineering. Some of the fundamental topic areas covered includes:

- combinatorial designs
- recurrence relations
- generating functions
- computational geometry
- graph theory
- enumeration trees
- abstract algebra
- logic and set theory
- data structures and algorithms
- discrete optimization

3 Learning Objectives

This course aims

- to fill the gaps in modern problems of Discrete mathematics;
- to learn practical problem-solving skills, which can be later applied in algorithmic theory;
- to develop fundamental knowledge of combinatorics and complexity;
- to develop practical skills needed in modern logic;
- to give practical knowledge, which is needed in many courses theoretical informatics.

4 Learning Outcomes

This course should help students to form the basic skills training to make and present their own research, motivate to engage in the scientific activity.

After completing the study of «Combinatorics, Graphs and Computational Logic» the student should:

1) *know basic concepts in combinatorics and group algebra, social network analysis, first-order logic and knowledge bases*;

2) *use combinatorial statements interpreted in generating functions of regular sets, use library Igraph for analyzing graphs in R programming environment, use Prolog for querying knowledge bases*;
3) *be able to analyze combinatorial problems, extract and interpret descriptive statistics from social networks, apply resolution techniques for finding the answer for forst-order query .*

5 Course Plan

№	Topic	Total hours	Contact hours		Self-study
			Lectures	Classes	
1.	Sets, relations, functions, simplest combinatorial formulas	20	2	2	16
2.	Euler formula for intersecting sets, Newton binomial, asymptotic combinatorial identities	12	2	2	8
3.	Linear recurrent sequences and regular generating functions	20	4	4	12
4.	Group action on finite sets	20	4	4	12
5.	Graphs and trees, basic theorems on graphs and coloring of graph	34	6	6	22
6.	Boolean logic and completeness of functional systems	34	6	6	22
7.	Predicate logic, basic notions, prefix form	16	4	4	8
8.	Normal forms, complexity for Boolean functions realizations by schemes and formulas	36	6	6	24
9.	Basis and finite total equivalence systems for closed classes of Boolean logic	36	6	6	24
Total		228	40	40	148

Some textbooks on discrete mathematics are written primarily for mathematics course, but the present course is intended for computing science students. The place of discrete mathematics in the undergraduate curriculum is now fairly well established, and it is certain that its place in the curriculum will be maintained in the third millennium.

Discrete mathematics has several aspects. One fundamental part is **combinatorics,** the study of counting arrangements of various types. We develop methods of counting which can deal with such problems. We begin from simple combinatorics and tend to recurrent sequences and action of a group on finite sets as a final resulting technique.

Next, **graph theory** can be used to model a variety of situations - road systems, chemical molecules, social networks, media interaction, etc. We introduce the basic types of graph and give some indication of what the important properties are that a graph might possess. We also describe boundaries for complexity of some basic algorithms.

At last, we deal with classic results in **computational logic** and its applications to many branches of mathematics. Such notions as completeness, finite generation, axiomatic completeness, and normal forms are considered. We also present a lot of counterexamples in many-valued logic and state open problems in this field.

Each topic ends with a good number of examples and final programming task. The examples are a mixture of fairly straightforward applications of the ideas of the topic and more challenging problems which are of interest in them or are of use later on in the course.

This course is new to HSE and Russian universities in general – it has some common points with the course of “Discrete mathematics”, which is read at department of Mechanics and Mathematics of Lomonosov Moscow State University, fourth course of specialist program. Modern tasks need strong perception and interpretation of problem on a pure mathematical language, clear understanding what we can do and what we can’t, and what are the conditions to optimize constructed algorithms and satisfy input and output limitations

6 Reading List

Recommended:

1. *R. Garnier, J. Taylor*. Discrete Mathematics for New Technology, Second Edition, IOP, 2002
2. Handbook of discrete and combinatorial mathematics / *Kenneth H. Rosen,* editor in chief, *John G. Michaels,* project editor, CRC Press, 2000.
3. *Kenneth P. Bogart, Scot Drysdale, and Cli Stein*, Discrete Math for Computer Science Students, 2004.
4. *O.B. Lupanov, A.B. Ugolnikov*. Selected papers.

In each part of a course three journal papers will be provided as a required reading on the current class subject.

Supplementary:

1. *Chen W.W.L.* Discrete mathematics, 1982.
2. *Andersen I.* A first course in discrete mathematics, Springer, 2002.
3. *L. Lov´ asz, K. Vesztergombi*. Discrete Mathematics Lecture Notes, Yale University, Spring 1999

7 Grading System

Control type	Assesntsme	Parameters
Current	Class work #1 (last week of third module) Class work #2 (last week of fourth module)	Written work during 120 minutes.
Final	Exam	Speaking exam with preparation time about 60 minutes

The assessment includes three main components:

- Class homework, assigned after each lecture
- Programs each month
- Final exam

Cumulative mark is average grade on homeworks and labs.

Final course mark *is obtained from the following formula*:

$$O_{final} = 0{,}6 \cdot O_{cumulative} + 0{,}4 \cdot O_{exam}$$

All these marks are evaluated using 10 grade scale. All grades having a fractional part equal or greater than 0.5 are rounded up.

Conversion of the concluding rounded grade to five-point scale grade is done in accordance with the following table:

Table of Grade Accordance

Ten-point Grading Scale	**Five-point Grading Scale**	
1 - very bad 2 – bad 3 – no pass	Unsatisfactory - 2	**FAIL**
4 – pass 5 – highly pass	Satisfactory – 3	**PASS**
6 – good 7 – very good	Good – 4	
8 – almost excellent 9 – excellent 10 – perfect	Excellent – 5	

8 Guidelines for Knowledge Assessment

Final exam questions

The final exam will consist of two questions and one task, with 60-minute time limit to prepare an answer and solve a task. All questions will cover all three chapters, spanning the entire scope of the course.

Topics for project and course work

- Many-valued logics “anti”-theorems and examples.
- Predicate approach in many-valued logics
- Labyrinth games

Typical class projects

-Group action on finite sets
-Algorithms on graphs and trees, problem of isomorphism
-Many-valued logics in comparison with Boolean logic

9 Methods of Instruction

The class is offered offline, with class material being 50% lectures and 50% classes.

Therefore, full student interaction with the tutor becomes the key to the class' success. The students should not be taking this course in theory-only mode, but have to verify their skills in programming real tasks.

We use a combination of teaching tools and methodology.

- Good theory representation with complete interactive experience.
- Constant interaction with the students.
- Class projects. While homeworks are meant to demonstrate the understanding of the current class material, we use major programming tasks for individuals and groups of students.
- Rich educational software support for all chapters.
- “Office hours”. They are usually scheduled on request, for one or many students, based on their current needs and questions.

10 Special Equipment and Software Support

Projector. Computer class with sufficient PC quantity for students.