Федеральное государственное автономное образовательное учреждение высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"

Факультет компьютерных наук
Департамент программной инженерии

Рабочая программа дисциплины
«Компьютерные сети»
(“Computer Networks”)
(на английском языке)

для образовательной программы «Программная инженерия»
направления подготовки 09.03.04 «Программная инженерия»
уровень - бакалавр

Разработчик программы
Р.А. Родригес Залепинос, к.т.н., доцент, rodriges@wikience.org

Одобрена на заседании департамента программной инженерии «___»_____ 2018 г.
Руководитель департамента Авдошин С.М. _____

Утверждена Академическим советом образовательной программы
«___»_____ 2018 г., № протокола_____

Академический руководитель образовательной программы
Шилов В.В. _____

Москва, 2018

1. Course Description

1.1. Course Title, Type and Pre-requisites

The elective course "Computer Networks" is offered to students of the Bachelor Program "Software Engineering", Computer Science Faculty (National Research University Higher School of Economics) during the second half of the 3rd academic year (quartiles 3 and 4, one semester).

Total credits – 5, hours in total – 190 (lectures – 32, practical lessons – 32, self-study – 126).

One lecture per week is planned which is accompanied with one practical lesson. The students are expected to devote about 8 hours per week for self-study.

The course has the following assessments: intermediate (one control work at the end of the 3rd quartile, two home assignments at both quartiles) and final (one exam at the end of the 4th quartile).

The mandatory prerequisites for this course are:

- Java programming skills obtained during the course "Software Design" provided for students of the Bachelor Program "Software Engineering" of the Computer Science Faculty during the 2nd academic year in two semesters (1 – 4 quartiles);
- Good knowledge of English language (at least listening and reading, intermediate speaking) since the course is given completely in English.

The experience in desktop, mobile or web development before entering the course is not required but will give more convenience during practical lessons and homework assignment selection.

1.2. Course Abstract

Computer Networks are ubiquitous in the modern IT sector. Computer Networks are arteries of databases, web & mobile applications, distributed systems and many other software. A good understanding of the underlying networking principles facilitates in building more reliable and efficient software products. This makes the Computer Networks course highly beneficial for any modern software engineer.

This course gives theoretical, practical and programming experience in the area of computer networks: how computer networks function, how do network protocols behave, what are the software to investigate computer networks, software networking frameworks and libraries to code networking applications.

In this course we will explore the core principles of computer networks. We will cover techniques for transmitting information efficiently and reliably over a variety of communication media. We will look at the addressing and routing problems that arise and must be solved during the information transmission. We will utilize different programming languages and popular technologies that make it possible to investigate the underlying principles of the network protocols (for example, Java and Netty networking framework). We will explore the TCP/IP stack including HTTP, FTP, and WebSocket.

Students will meet weekly with the instructor (1 lecture and 1 practical lesson weekly). The course is taught as a hands-on training when lecture material is covered interactively together with the course instructor. The course contains programming assignments as well as network exploration using the respective software tools.

This course includes 2 home assignments.
This course also has 1 control work (in the middle of the course) and 1 final exam.

1.3. Course Site

The course site will be specially set up to make lecture slides, source code for practical lessons, links to software tools and libraries, correct answers to control work and exam questions publicly available. The goal for this is twofold. First, maintaining high-quality course material on-line allows both to contribute to community resources and HSE reputation. Second, a student may add the course to their CV and use its material as a handbook later in future. The course trainer guarantees to keep up the course site as long as at is reasonably possible.

1.4. Course Syllabus

This syllabus describes topics to be covered during the course, knowledge and skills that a student will gain after successful accomplishment of the course. It also regulates the assessment types and criteria for marking. The course embraces wide range of modern technologies in the world of computer networks including widely used networking protocols, tools, and software frameworks/libraries. A considerable portion of the course is devoted to the TCP/IP which is one of the most popular nowadays and gaining networking programming experience.

The syllabus is prepared for teachers responsible for the course (or closely related disciplines), teaching assistants, students enrolled on the course as well as experts and statutory bodies carrying out assigned or regular accreditations in accordance with

- Educational standards of the National Research University "Higher School of Economics" (HSE);
- Bachelor Program Curriculum "Software Engineering" (area code 09.03.04), 3rd year, 2018–2019 academic year;

2. Learning Objectives

Upon the accomplishment of this course, a successful student should be able to understand basic networking principles, know the functioning of popular networking protocols, be able to programmatically develop a networking application that uses popular network protocols.

3. Learning Outcomes

After completing “Computer networks” course, a successful student will:

- Have strong understanding of networking concepts and computer networks functioning;
- Be able to programmatically work with popular computer networking protocols;
- Acquire skills and experience in modern technologies and tools related to computer networks.

A successful student should:

Know:

- Core concepts of computer networks;
- Message formats of the most common network protocols;
- Popular software tools that are able to monitor computer networks;
- Most widely used networking services;
- Modern networking frameworks and/or libraries.

Be able to:

- Interpret messages of the most common network protocols that were captured by popular software tools for network monitoring;

- Programmatically work with popular network protocols using modern networking frameworks/libraries;
- Design custom network protocols on top of existing network protocols;
- Build basic networks in a virtual network simulator.

Acquire experience in:

- Programmatic work with diverse computer network protocols;
- Use software libraries and tools to programmatically/manually manage computer networks.

This course forms the competencies (according to the educational standards of National Research University Higher Schools of Economics – 09.03.04 “Software Engineering”, Bachelor level, protocol approved on 14.06.2017, №2) described in the table below. More detailed description of the theory and practice that form the listed competencies as well as evaluation criteria are given in the corresponding sections of this syllabus (Course Plan and Course Assessment).

Competencies marked with * (asterisk) are optional and are formed in the case when a student chooses an optional activity (see “Alternative ways of assessments” section in this syllabus). The experience from previous academic years and courses built using the same strategy (for example, course “Geoapplications development” http://rgeo.wikience.org/) shows that over 80% of students form teams expressing the will to acquire optional competencies.

Competency code	**Competency description**	**The way in which “Computer networks” course forms the competency**
	I. Universal competencies	
УК-1, СК-Б1	Able to learn, acquire new knowledge, skills, including in the area other than the professional	In addition to other material, this course gives knowledge of some basic physical principles on which computer networks interconnection is built (for example, optic fiber and other communication media that are essential to computer networks).
УК-3, СК-Б4	Able to solve problems in professional activities on the basis of the analysis and synthesis	This course is divided onto sections covering a wide variety of protocols, tools, hardware, frameworks, network topologies for building real-world computer networks. These provide “bricks” that should be used for designing a basic computer network given certain requirements or developing a respective networking software (more focus is on the software as this course is not about network administration). The aforementioned goals could be achieved by analyzing and dividing the given complex task (e.g. homework assignment) onto a sequence of simpler tasks.
УК-5, СК-Б6	Able to work with the information: locate, evaluate and use information from different sources necessary for solving scientific and professional problems (including on the basis of the system approach)	To successfully complete homework assignments and work during practical lessons, students have to find additional information about the networking tools, protocols, and technologies which they will use for their tasks. They also need to find the appropriate ways (usually check the standard specification, look through the documentation, etc.) to reach the specified goals.
УК-6, СК-Б7 *	Able to carry out research, including	When a student chooses to do a research (see “Alternative ways of assessments” section in this syl-

Competency code	Competency description	The way in which “Computer networks” course forms the competency
	the analysis of the lems, setting goals and objectives, selecting object and subject of the study, choose research methods, and assess quality of the resulting research	labus).
УК-7, СК-Б8 *	Able to work in a team	Teamwork is highly encouraged to complete the final homework assignment but is not mandatory. It is expected that about 50% of students or more will form teams.
II. Professional competencies		
Research activity		
ПК-1, ИК-1 *	Able to apply the basic concepts, principles, theories and facts related to computer science to solve research problems	When a student chooses to do a research (see “Alternative ways of assessments” section in this syllabus).
ПК-5, ИК-5 *	Able to prepare presentations, scientific and technical reports on the results of the work performed, publish research results in the form of articles/reports/papers on scientific conferences	When a student chooses to do a research (see “Alternative ways of assessments” section in this syllabus).
Analytical activity		
ПК-6, ИК-6	Able to formalize the subject area of software project and develop specifications for software components	Homework assignments involve programmatic work with computer networks, protocols, frameworks and tools. This requires understanding and formalizing the subject area.
ПК-8, ИК-8 *	Able to prepare commercial offers with solution versions	Students are encouraged to form a startup which involves developing a software that utilizes computer networks (see “Alternative ways of assessments” section in this syllabus).
Project activity:		
ПК-9, ИК-9	Able to create software for computers and systems of different architecture	The main cross-platform technologies used in this course are Java, JavaScript, Python. Software built on those runs on diverse platforms without modifications in most cases (MacOS, Linux, Windows). Networking technologies that are covered within this course are also platform interoperable. Students are also encouraged to choose homework projects for Web and Mobile platforms.
ПК-10, ИК-10	Able to design, construct and test software	Homework assignments should meet careful design and testing state-of-the-art criteria. Test-driven approach via JUnit or TestNg frameworks or other (depending on the language) for quality assurance (QA) should be exploited.
ПК-11, ИК-11	Able to read, understand and highlight the main idea ter reading source code and	Since the course involves programming and mastering new software libraries, some exam and control work questions are based on the knowledge of

Competency code	Competency description	The way in which “Computer networks” course forms the competency
	software documentation	source code and documentation of the software tools covered during the course.
Technological activity		
ПК-15, ИК-15	Able to use operating systems, network technology, software interface development tools, use languages and methods for formal specification, database management systems	– this course is itself targeted at learning diverse networking technologies – students gain experience in generic high-performance network frameworks (they are not limited to a particular type of data or a particular purpose): Apache Mina, Netty, Java Sockets – students will gain some experience in Linux and cross-platform tools for Linux/Windows/MacOS
ПК-16, ИК-16	Able to use diverse software development technologies	Students use during the course (not limited to): object-oriented design, test-driven development, version control systems, Web and Mobile development technologies (depending on the selected topic) and other technologies.
Development activity:		
ПК-17, ИК-17	Able to use basic techniques and software development tools	Students use the following techniques and software development tools for their practical lessons and homework assignments (incomplete list): * Rapid Application Development (RAD): IntelliJ IDEA or Eclipse; * Project build managers: Maven/Gradle.

4. Course Plan

4.1. Course Outline

Students should devote self-study hours to learning material for self-study (about 15–35% of the material from each topic), doing their homework, control work, and exam preparation.

№	Topic name		Course hours, total	Audience hours		Self-study
				Lectures	Practical lessons	
Module 3						
1	Introduction to Computer Networks		4	2	2	0
2	TCP/IP Stack	TCP/IP stack: overview, addressing mechanisms, WireShark tool	8	4	4	15
3		DNS (Domain Name Service)	4	2	2	10
4		IP (Internet Protocol)	8	4	4	15
5		TCP (Transmission Control Protocol)	8	4	4	20
Subtotal:			**32**	**16**	**16**	**60**
Module 4						
6	Routing protocols		4	2	2	8
7	Netty Framework		8	4	4	18
8	Creating Computer Networks in Microsoft Azure Cloud		4	2	2	12
9	Networking Services		4	2	2	10
10	Computer Networks Security		8	4	4	15
11	Recent trends in Computer Networks		4	2	2	5
Subtotal:			**32**	**16**	**16**	**66**
Total:			**64**	**32**	**32**	**126**

Lectures will be given in the form of PowerPoint presentations. Lecture slides may be made publicly available at the course site (site address will be announced during the beginning of the course). The slides will contain more links to additional Web resources (readings, documentation, tools). Students who experience difficulties in watching the presentation from laser projector may download the slides onto their own laptops or a class PC to be able to comfortably track the lecture.

Information about the data required for practical lessons, software tools (links to web sites) and libraries (Maven dependencies to be added to project's pom.xml) may be published at the course site (site address will be announced during the beginning of the course). Students are encouraged to install software tools, resolve Maven dependencies for libraries and download all necessary data before attending classes to save time. Lecturer reserves the right to notify students via e-mail about the material necessary for the next classes or changes in the schedule and/or material.

The source code in Java and other languages that is demonstrated during practical lessons will be also publicly available at the course web site for each lesson (site address will be announced during the beginning of the course).

4.2. Topic-wise Course Content

Topic 1. Introduction to Computer Networks.
A large fraction of modern software is based on computer networks, history of computer networks, nowadays networking, 100GB/s network speed and beyond, readings, RFC (Request For Comments). MAC addresses, the structure of MAC address. Address assignment regulations, IEEE Registration Authority. Computer Networks Simulators.

Topic 2. TCP/IP stack: overview, addressing mechanisms, WireShark tool.
The stack structure, the correspondence of the TCP/IP stack layers to the OSI Model layers. The review of the TCP/IP stack protocols (details on these protocols are given in the respective parts of the course): IP, ICMP, RIP, OSPF, TCP, UDP, FTP, Telnet, HTTP, SMTP, SNMP, TFTP. Data entity classification in the TCP/IP protocol stack: stream, datagram, segment, packet, frame.

Address types of the TCP/IP Protocol Stack: local (hardware) addresses, network IP addresses. An IP address is an address of a network interface, not a node (network card, router interfaces, etc.). Domain names, basic introduction to DNS.

IP Address Formats. Network address and host address. IP address masks. Classes of IP addresses.
Destination address types: unidentified, unicast, multicast, limited broadcast, directed broadcast, broadcast, anycast (IPv6), loopback (127.*.*.*).

Centralized assignment of IP addresses: ICANN (Internet Corporation for Assigned Names and Numbers), the deficit of IPv4 addresses. CIDR (Classless Inter-Domain Routing).

ARP (Address Resolution Protocol), RFC 1122. ARP messages (requests, responses, gratuitous messages, targeted ARP requests), ARP message formats. ARP server, ARP tables, static and dynamic ARP records, time-to-live of dynamic records, ARP cache. RARP (Reverse ARP). Proxy-ARP.

Practical lesson. Investigation of ARP activity over the network.

- Tools:
  - `arp` (command line, both Windows and Linux)
  - `ipconfig/ifconfig` (command line, Windows/Linux)
  - WireShark (GUI, both Windows and Linux, many other platforms are also supported)
- Read more about “Promiscuous mode” at the WireShark Wiki: https://wiki.wireshark.org/CaptureSetup/Ethernet

Topic 3. DNS (Domain Name Service).
Hierarchical name space, domains, subdomains. Short, relative, complete domain names. ICANN (Internet Corporation for Assigned Names and Numbers). ISO 3166, country domains. File hosts.txt. DNS protocol and DNS servers. DNS zones, authoritative and non-authoritative responses. DNS poisoning.

Types of DNS records: Address Mapping records (A), IP Version 6 Address records (AAAA), Canonical Name records (CNAME), Host Information records (HINFO), Integrated Services Digital Network records (ISDN), Mail exchanger record (MX), Name Server records (NS), Reverse-lookup Pointer records (PTR), Start of Authority records (SOA), Text records (TXT).

DNS name resolving: recursive and non-recursive procedures. Root DNS servers. The use of anycast technique. Reverse DNS zones.

Topic 4. IP (Internet Protocol).
IP purpose. IP packet. The header of IP packet. TTL (Time To Live). Checksum. IP routing scheme. Routing tables. Route types. Algorithms for traversing routing tables. IP routing with and without masks.

IP packet fragmentation. Fragmentation parameters. Fragmentation mechanism. ICMP (Internet Control Message Protocol). The format of ICMP messages. `traceroute`/`tracert` utility. `ping` utility.

IPv6. IPv6 addressing system. The structure of global aggregating unique IPv6 address. Decreasing the load on routers. Switching to IPv6.

Topic 5. TCP (Transmission Control Protocol).
Multiplexing and demultiplexing. TCP ports. TCP sockets. UDP (User Datagram Protocol). Stateful and stateless protocols. UDP datagrams. Dataflow scheme in TCP and UDP. TCP segments. ACK notion. TCP segments dataflow scheme. The format of TCP header. TCP logical connections. TCP connection scheme. Automatic repeat request methods: source idle, sliding window, N-segments back. TCP buffer structure. Accumulated ACK principle. Dataflow control in TCP.

Topic 5. Routing protocols.
Common properties and classification of routing protocols. Source routing. Static and adaptive routing. Distance Vector Algorithms (DVA), Link-State Advertisement (LSA).

RIP (Routing Information Protocol). Building routing table. Router adaptation to network state dynamics. False routes.

OSPF (Open Shortest Path First). Two stages of building routing table. Metrics. Routing in heterogenous networks. Using several routing protocols in the same network.

IGMP (Internet Group Management Protocol). The structure of IGMP Message. Routing principles.

Topic 6. Netty Framework.
Architecture of Netty. Java InetAddress, blocking and non-blocking sockets, java.nio. Bootstrap or ServerBootstrap, EventLoop, EventLoopGroup, ChannelPipeline, Channel, Future or ChannelFuture, ChannelInitializer, ChannelHandler. Building custom multiuser server. Creating TCP and HTTP servers. Customizing business logic. Designing custom networking protocols on top of TCP and/or HTTP.

Topic 7. Creating Computer Networks in Microsoft Azure Cloud.
The description and peculiarities of Microsoft Azure Cloud. Java Azure API. Web GUI Azure console. Virtual machines, network interfaces, network security groups, network routing rules, load balancers. API keys and access setup to Azure Cloud. Building a computer cluster in Microsoft Azure Cloud: a set of virtual machines connected with a single network. Choosing computer network size and selecting address masks. Creating and assigning IP addresses. Using a single IP address and different TCP ports to access different virtual machines. Monitoring the network activity in Microsoft Azure Cloud.

Topic 8. Networking Services.
Web servers. HTTP (HyperText Transfer Protocol). URL (Uniform Resource Locator). Web client and web server. The format of HTTP messages and headers. Investigating HTTP using Netty framework. WebSocket protocol. Binary and text modes. Creating WebSocket client and WebSocket server.

Mail servers. Electronic messages. Dedicated mail servers. Overview of POP3 (Post Office Protocol Version 3), SMTP (Simple Mail Transfer Protocol), and IMAP (Internet Message Access Protocol) mail protocols.

File servers. Overview of FTP (File Transmission Protocol) and Amazon S3 (Amazon Simple Storage Service). Caching, replication, fault-tolerance. File service in the Cloud. Problems and solutions for transmission of large files.

Topic 9. Computer Networks Security.
Traffic filtering, filtering rules. Firewalls. Software and hardware firewalls. Proxy-servers. NAT (Network Address Translation), NAPT (Network Address Port Translation), DMZ (Demilitarized Zone), IDS (Intrusion Detection Systems). Overview of IPsec and VPN (Virtual Private Networks). Overview of SSL (Secure Sockets Layer).

Types and classification of computer network threats and attacks. TCP attacks, ICMP attacks, UDP-attacks, IP attacks. DoS (Denial of Service), DDoS (Distributed DoS), Man in the Middle, DNS attacks (DNS Poisoning, DNS Spoofing, Attacks on DNS Root Servers). Malware: trojans, worms, viruses, botnets. Methods of computer networks protection.

Topic 10. Recent trends in Computer Networks.
Large Modern Computer Networks. Grid systems. Large computer clusters. In-memory systems and networking issues. Networking hardware becomes smarter: solutions of Intel, Cisco, Mellanox and other companies for increasing the performance and value of networking hardware.

4.3. Topics for control work and final assessment

Any question targeted to check the understanding of any topic listed in “Detailed Course Content” section of this syllabus may be asked during the control work or the final exam. To answer the questions successfully, a student will need programming experience obtained during practical lessons in addition to lecture material.

Before control work or exam, up to 20 minutes at a lecture or practical lesson will be devoted to “Test Preparation”. This preparation is held in the form of joint (all students involved) answering (orally) sample questions similar to control work/exam questions.

Test preparation, control work, and exam formats are the same as standard Software Development certification programs, English tests or similar certification. You may find the examples for the “Geoapplications development” course below. The format for “Computer Networks” course is the same.

Please, find a test preparation example at:
http://rgeo.wikience.org/pdf/tests/rgeo-course-TestPrepare01.pdf

Please, find a control work/exam example at (correct answers are **in bold**):
http://rgeo.wikience.org/pdf/tests/rgeo-course-exam_example.pdf

It is highly recommended for a student to revise material using lecture slides and source code from practical lessons before they come to control work/exam.

It is also recommended to download all the lecture slides again before revision (from the course site), even if a student has downloaded all of them previously (e.g. each time classes start). This is because after the lecture the instructor reserves the right to make modifications to slides including the correction of possible mistakes (if any) and addition of answers to questions asked by students during the lecture or practical lesson to increase their understanding of the material and help them to prepare to the control work/exam.

4.4. Topics for homework assignments

4.4.1. Homework assignment 1

This assignment consists of a series of micro-homework. It has several types of assignments. Students will receive a task on which they start working in class and finish the task at home. The tasks and requirements for them may be partially formulated in oral form by the instructor.

One type of assignments is targeted at the investigation of the protocol structure using popular networking tools. For example, students may use WireShark to capture network traffic and study the format of protocol headers.

The other type of assignments is the design of a virtual network using a network simulator (if it will be freely available to the students and the lecturer). A student will construct a computer network consisting of routers, nodes, servers, and other components. This will enable a comprehensive understanding of networking by students.

4.4.2. Homework assignment 2

In addition to the topics listed, in the beginning of some lectures, up to 15 minutes will be devoted to "Networking surveys". They are held in the form of PowerPoint presentations by the course instructor. Their goal is to present the students the wealth of diverse interesting networking problems and tasks that could be solved using networking technologies.

The second homework is devoted to developing a software networking application using a networking framework, e.g. Netty. A student should develop an HTTP/TCP server, custom protocol, or something related to a topic of our course.

Before a student starts doing their homework, they have to arrive at a task to be solved with the application they aim to develop. Networking technologies, software, tools, frameworks must be central to their work. However, since the area of computer networks is new to students, they are not aware of the interesting challenges. The surveys are held to give examples of both problems and technologies.

Homework topics are not limited to the tasks to be described in the surveys. Students may also seek networking technologies and interesting tasks to solve for their homework assignments on their own.

Students are encouraged to form teams or choose working on a startup, or just interesting projects or research projects related to Computer Networks. Please, see "Alternative ways of assessment".

The topic for home assignment may be proposed by the course instructor, student (should be approved by the instructor), or worked out jointly with the instructor and a student (or group of students in case of teamwork).

In any case, before working on a chosen topic, students should have their decision approved by the course instructor. The instructor will give a deadline before a topic must be approved. The failure to approve the topic before the deadline results in 0 mark for the home assignment. The instructor may modify the task and/or data by some considerations including if he/she feels them unrealistic or too easy to implement. The final decision on which data and tasks are right for the homework is left to the sole discretion of the instructor.

In this course, students are highly encouraged to develop a complete and quality-assured geoapplication that can be useful to others, published on the Web, and/or demonstrated at programming competitions and/or personal portfolios.

4. Reading list

This section contains core required and optional reading for this course. Please, refer to “Course Plan” section of this syllabus which has detailed readings lists for each topic.

4.2. Required

- Victor Olifer and Natalia Olifer, Computer Networks: Principles, Technologies and Protocols, 5th edition, Piter, 992 P., 2017.
- James Kurose, Keith Ross, Computer Networking: A Top-Down Approach, Pearson, 7th edition, 864 P., 2016.
- Andrew S. Tanenbaum, David J. Wetherall, Computer Networks, 5th edition, Prentice Hall, 960 P., 2012.
- Christian Benvenuti, Understanding Linux Network Internals, O’Reilly Media, 1064 P., 2006.
- N. Maurer, Netty in Action, http://topconsulting.ru/wp-content/uploads/2011/03/Netty-In-Action-V5.pdf (this book is freely available on-line)

4.2. Optional

- Steven Noble, Building Modern Networks: Create and manage cutting-edge networks and services, Packt Publishing, 324 P., 2017.
- Jose Manuel Ortega, Mastering Python for Networking and Security: Leverage Python scripts and libraries to overcome networking and security issues, Packt Publishing, 426 P., 2018.
- James Bernstein, Networking Made Easy: Get Yourself Connected (Computers Made Easy), Independently published, 141 P., 2018.
- Andrew Crouthamel, Mastering Wireshark 2: Develop skills for network analysis and address a wide range of information security threats, Packt Publishing, 326 P., 2018.
- Netty short guide, http://docs.jboss.org/netty/3.2/guide/pdf/netty.pdf

5. Grading system

The 10-points scale is basic for all types of assessment. The contribution of each activity to the final grade is given in table below.

Activity Contribution Table

Code	Description	Contribution, %
LP	Lecture presence	20
PP	Work during practical lessons	10
CW	Control work	10
HA1	Home assignment 1	25
HA2	Home assignment 2	15
EX	Exam	20
TT	**Total contribution**	

The final grade is calculated according to TT:

Final Grade Calculation Table

Total contribution, %	Final grade
<= 10	0
<=15	1
20	2
30	3
40	4
50	5
60	6
70	7
80	8
90	9
over 95	10

LP and PP are simply the percent of classroom attendance by student expressed in percent (0% never attended, 100% attended all classes).

HA1 percent of accomplishment is calculated as HA1PA = (T / N), where N is the total number of micro-assignments for HA1, and T is the sum of all assignments' marks that were evaluated higher than 5 by the instructor (10-points scale). HA1 = HA1PA * 2.5. The instructor gives 0 points for a micro-assignment that was not shown.

Please note, that the homework assignments should be defended during a practical lesson within a defined deadline by the instructor (please, see the respective section of this syllabus for more details).

Final grade conversion rules from 10-levels grade to 5-points grade is given in table below:

Conversion between 10-levels and 5-levels grading system

10-points scale	5-points scale
1 – unsatisfactory 2 – very bad 3 – bad	unsatisfactory – 2
4 – satisfactory 5 – quite satisfactory	satisfactory – 3
6 – good 7 – very good	good – 4
8 – almost excellent 9 – excellent 10 – brilliant	excellent – 5

If a student is late for a lecture or a practical lesson for more than 20 minutes, they will be considered absent. It is prohibited to visit classes wearing shorts; a student will be considered absent in the case of violation of this rule; in addition, it will be impossible to defend a home assignment. A home assignment cannot be defended during a break between classes. It is prohibited to use computers, mobile phones, Internet and anything else during lectures for not viewing lecture slides or during practical lessons for not practicing the given assignment. The instructor will penalize students violating this requirement by subtracting scores from LP and PP. The final mark is rounded upwards. Other marks are being kept precise during the whole course.

6. Guidelines for Knowledge Assessment

6.1. Assessment types and forms

This course has two types of assessment (intermediate and final) with four forms of assessment (one control work, two home assignments, and one final exam). All types and forms of assessment have influence on the final grade. The first and the only control work will be held at the end of the first course module (quartile). The home assignments assume that a student will complete the given task based on the knowledge and experience gained during this course. The first home assignment is for the first quartile quartile while the second home assignment is planned on the second module (quartile) and results in a software networking application. The final exam is planned on the end of the last module (quartile).

The table below summarizes assessment types and forms.

Assessment Table: Assessment types and forms in quartiles (modules)

Assessment type	Assessment form	Quartiles (modules)		Notes
		1	2	
Intermediate	Control work (test)	*		TCP/IP Stack Protocols
	Home assignment 1 (defense on practical lessons)	*		Understanding networking protocols, principles, and functioning
	Home assignment 2 (software development and defense)		*	Develop a networking application that uses software networking technologies/frameworks
Final	Exam (test)		*	All course material

All results are evaluated according to 10-level scale from 0 (failure) to 10 (excellent) inclusively.

6.2. Control work and exam forms

The control work and exam will be in the form of written tests (1 hour 20 minutes each). They will be designed using state-of-the-art understanding of how a good test should look like. The tests that will be provided are similar to Cambridge English Certification and Oracle Java Certified Programmer exam.

The control work and exam will be held in written form. Printed forms with questions and answer versions will be given to students. Questions and answers are in English. Students are not required to write any answers, correct answer(s) should be circled with a pen. Corrections are not allowed (so you can answer the questions using a pencil first or draft sheets). Draft sheets are the opposite side of the exam forms. A student should have a pen with them. The absence of a pen will be considered as control work/exam failure.

Cheating: use of mobile phones, Internet, books, notebooks is prohibited during control work/exam. Communication with other students is prohibited (even asking a neighbor for today's date). Violation of the rules will be penalized.

To avoid confusion, students should provide their surname, name, patronymic, current date, group and signature on the first exam sheet. Repeating names on each exam sheet is not required. Each document has a unique generated ID in its footnotes (e.g. Random Document ID: ndzoyCaK). This is achieved by VBA macro for MS Word that is run before printing each exam sheet.

Evaluation criteria for questions are as follows. Correct answer to a question gives **1.0** points. Answers to questions with multiple choices will be evaluated as follows:

- All correct items were checked without any incorrect ones — 1.0 (full score)
- At least one correct item was checked without any incorrect ones — 0.5 (half of a score)
- At least one incorrect item was checked — 0.0 (no points)

No total points (0 score) gives 0% contribution to the final mark, all correct answers to questions give 20% contribution to the final mark (see corresponding syllabus section for details).

The same format is used in the "Geoapplications development" course. Please, find control work/exam example from that course at (correct answers are **in bold**):

http://rgeo.wikience.org/pdf/tests/rgeo-course-exam_example.pdf

6.3. Homework assignment form and assessment

6.3.1. Homework assignment 1

This kind of homework ensures regular practical experience of students at home. After the material for a topic is covered by lecture, students receive a series of small tasks for the homework (micro assignments). Students should hand-in a micro-assignment in 1 or 2 weeks after it was assigned (exact deadline will be noted by the instructor for each assignment separately in oral or written form).

The instructor will check the homework only during practical lessons. Thus, absence on a practical lesson devoted to homework assignment control is considered to be a homework delay. The instructor penalizes delays by subtracting scores from the final mark of given homework. Student may earn 0 – 10 points for a homework.

In general, the criteria for homework assignments is the understanding of the networking protocols, concepts and network functioning. The teacher will use traditional technique to check this: they will ask questions during homework defense to ensure understanding the material by student, source code written and source code authenticity (if any). Questions may be also based on the lecture material. The teacher marks the work according to the percent of questions answered, the amount of work done, the volume of work done, accuracy of source code and the application as a whole (if any), correctness of the application (if any) and other sound criteria that are applicable to this kind of work.

These homework assignments (not homework assignments № 2) should be done individually by each student. Cheating including code borrowing or its other forms will be penalized down to earning 0 scores for the assignment.

6.3.2. Homework assignment 2

The homework defense will be held in the form of presentation of the developed geoapplication to the groupmates. Each student must submit PowerPoint presentation, complete source codes and architecture description (Microsoft Word document) of their networking application for teacher's review. Once all critical comments are fixed, a student submits their work for group review in a dedicated forum topic. Other group members should ask at least 1 question clarifying the application goals, its internal structure and usage. The activity of asking questions will also contribute to the homework mark. The questions should be constructive; the answers for them may not be evident. The examined student must provide answers to all questions related to home assignment. Based on the gathered questions the student will have to refine their presentation and architecture description. After discussion and refinement are complete, the student submits the final version of their application. The developed application should be covered with unit tests via JUnit, TestNg or other framework for quality assurance. Test-driven development (TDD) approach should be applied during the application development. The quality of tests and coverage percent influence the homework mark. The mark for the application is given in the accordance to state-of-the-art evaluation criteria common to all courses that require application development during home assignments or other similar assessment activity. The mark for homework will

also depend on the variety of networking technologies exploited in the application. Tentative topics for home assignments are given in section ”Topics for homework assignments”.

6.4. Alternative ways of assessments

During the course a student might want to substitute Home Assignment and Exam (and possibly some practical lessons) with other types of activities.

A student (or team of students) may propose an application project to work on, for ple http://www.wikience.org/ru/темы_проектов/исследуй-климат-сам/. This will result in complex programming experience (teamwork, development of a complete software product, etc.). The resulting project may be presented at corresponding competitions (e.g. IBM Smarter Planet, Microsoft Climate Initiative, Google Earth Engine Awards) or commercialized.

The other way is to tackle a research problem agreed with the course instructor in advance. This includes reading research papers, developing new methods/algorithms, publishing journal or conference research paper(s).

6.5. Exam retake commission

The format of the exam retake is the same as during an ordinary exam format for this course: please, refer to Section 6.2 for details. A student will not be asked any additional questions in oral, written or any other form. The cumulative mark for a student (all items from Section 5 except “EX”) may be considered by the exam retake commission. The commission may use the formula for calculating the mark which can be found in Section 6. Item “EX” means the mark that was earned by the student during the exam retake (note that exam retake mark may contribute to the final course mark with weight 0.2).

7. Methods of Instruction

The considerable part of the course is held in the form of hands on training. Quite often students are expected to try out things covered on slides together with the trainer during a lesson (4 academic hours in class planned for 1 lecture and 1 practical lesson).

Theoretical material on the slides is largely interleaved with practice. Lectures have forms of PowerPoint presentations (during some lectures students are expected to work with their computers). Practical lessons are provided as trainings: instructor defines tasks to be accomplished using a programming language, networking monitoring tools or other technologies (according to the current topic).

Instructor helps to overcome difficulties that arise during the implementation of the given tasks by asking leading questions, pointing to corresponding Web resources, documentation. They also suggest improvements to the code or other results from the assignment.

8. Special Equipment and Software Support

All learning resources are freely accessible via Internet. The respective links are given to students. Course instructors do not issue directly to students any software, books, documentation or other material even if they are freely available on the Web.

8.1. Software support

Below is the main list of software that will be used during the course:

- Java Development Kit (JDK) at least 1.8 version
- WireShark

- Windows or Linux
- arp, ping, tracert
- IntelliJ IDEA or other IDE
- Netty framework

8.2. Remote support

The course site will be established with all lecture slides, source code, links to resources and data that are publicly available, and, possibly, LMS (notification of students by e-mail, control work/exam results publication, gathering homework assignments) are used for remote course support.

Students may also use the instructor's institutional e-mail arodriges@hse.ru to contact him directly. It is also possible to arrange Skype meeting with the instructor (please, contact him via e-mail to schedule a meeting).

8.3. Technical resources

Students should prefer to take their personal laptops for each practical lesson. This will accelerate their learning and reduce stress since they will be working in familiar and highly personalized environment.

The lecturer uses laptop and laser projector for presentations and practical lessons.

Students use Java IDEs like IntelliJ IDEA or Eclipse. In addition, they will use various tools and technological for network monitoring, configuration, design, and other necessary activities. The concrete software packages and tools that are required to accomplish a given practical task are named at the respective practical lessons.

The author of the program: / R.A. Rodriges Zalipynis /