International Information Technology University JSC

Faculty of Information Technology

Department of Computer Engineering and Information Security

Approved
Vice-Rector for Academic and
Educational Affairs of IITU JSC, PhD
_____ Umarov T.F.
«___» _____ 20__

SYLLABUS
(ACADEMIC PROGRAM)

Course: AiSD 2216 Algorithms and Data Structures (SDP 4)

Major: 5B070400 Computer Systems and Software Engineering

Year: 2, **Semester**: 4; **Number of credits**: 5 ECTS

Lectures: 15 hours

Laboratory classes: 15 hours

Practical classes: 15 hours

T/SIS: 105 hours

Total: 150 hours

Final assessment form: Final project

Almaty 20__

31.01.20

Academic Program of the course « AiSD 2216 Algorithms and Data Structures (SDP 4)» has been developed on the basis of a Model Academic Program.

Academic program has been reviewed at the meeting of «Computer Engineering and Informatioin Security» department.

Minutes № _____ dated «____» _____ 20___

Head of the Department _____ PhD, assoc. prof. N.T. Duzbayev

Author _____ MSc, senior-lecturer Tolegenov A.M.

The working academic program has been approved at the meeting of Educational and Methodological Board of JSC "IITU"

Minutes № _____ dated «___» _____ 20___

Director of the Department
for Academic Affairs _____ A.K. Mustafina
signature

1. GENERAL INFORMATION	
Faculty	Information Technology
Major code and title	5B70400 Computer Systems and Software Engineering
Year, semester	2 year, 4 semester
Subject category	Elective
Number of Credits	5 ECTS
Language of Delivery	English
Prerequisites	Application Development in Java (SDP 2)
Postrequisites	Java Enterprise Edition
Lecturer	Tolegenov A.M, senior-lecturer., master of tech. sc., azicus.sdu@gmail.com office 409 office 409 Office hours Monday-Friday 13.00-15.00
Lab and practice assistants	Karabaliyev Y.N., lecturer., master of tech. sc., bayern_707@mail.ru office 409 Balgabek A.A., lecturer., master of tech. sc., askar.balgabek@gmail.com office 409

2. GOAL, OBJECTIVES AND LEARNING OUTCOMES OF THE COURSE

The **goal** of the course is to learn basics of Object-Oriented Programming by using Java SE language. The **objectives** of the course are to:

- Teach students to implement simple and recursive algorithms based on data structures
- learn main concepts of data structures
- Acquire advanced principles of Object-oriented programming using data structures
- Learn to create desktop application using algorithms and structures of data

Learning outcomes of the course

Students successfully completing this course will be able to:

- Implementing own algorithms based on tasks
- Understand:
  - The basics principles of Object Oriented Programming in data structures
  - Linear and Tree based data structures
  - Advanced sorting and graph algorithms
  - Implementing Graphs

Develop standalone desktop applications by using design patterns

3. COURSE DESCRIPTION

Course goals is addressed to good principles of algorithm design, algorithms analysis, and foundational data structures. The emphasis is on selecting appropriate data structures and designing efficient and correct algorithms in execution of these data structures.

Object Oriented Programming, searching and sorting advanced algorithms, linear and tree based data structures, graphs.

Student will be able to practice on real desktop standalone projects and tasks. In addition, student will be able to study new trend technologies by researching.

4. COURSE POLICY

Students are not allowed to miss classes and use cell phones during classes. There are no additional lessons or workshops for students who missed lecture or laboratory class. Late submission of written assignments will be penalized 2% (minus two percent) for every day late. Students may come to see Tutor only at Office Hour's time or by appointment. Deadline for each task must 1 week.

Attendance/participation is assessed based on the number of lessons attended and participated in by students. Each student has to visit 80% of lessons differently he will not be allowed to final exam. After of absent of lesson he should present for teacher permission for access to lessons from dean.

5. LITERATURE

Basic literature:

1. Introduction to Java programming and Data Structures, Y. Daniel Liang. – Pearson 2018
2. Problem solving in Data Structures and Algorithms Using Java, Hemant Jain - 2016
3. Data Structures and algorithms Made in Java, Narasimbha Karumanchi – CarrerMonk 2017

Supplementary literature:

4. https://www.geeksforgeeks.org/data-structures/
5. https://www.tutorialspoint.com/data_structures_algorithms

6. Course schedule

Week date	Lectures	References	Lectures	Practice Classes (1h/w)	Lab. Sessions (1 h/w)	TSIS (2 h/w)	SIS (4 h/w)
1	Big-O notation, recursion	[1-2]	L1 (1 hour)	PW 1 (1 hour)	LW 1 (2 hour)	TSIS 1 (1 hour)	SIS 1 (6 hours)
2	Basic sorting algorithms	[3-4]	L2 (1 hour)	PW 2 (1 hour)	LW 2 (1 hour)	TSIS 2 (1 hour)	SIS 2 (6 hours)
3	Advanced sorting algorithms	[1-3]	L3 (1 hour)	PW 3 (1 hour)	LW 3 (1 hour)	TSIS 3 (1 hour)	SIS 3 (6 hours)
4	Generics	[3-4]	L4 (1 hour)	PW 4 (1 hour)	LW 4 (1 hour)	TSIS 4 (1 hour)	SIS4 (6 hours)
5	Array, linked lists	[2-4]	L5 (1 hour)	PW 5 (1 hour)	LW 5 (1 hour)	TSIS 5 (1 hour)	SIS 5 (6 hours)
6	Queues and priority queues	[4]	L6 (1 hour)	PW 6 (1 hour)	LW 6 (1 hour)	TSIS 6 (1 hour)	SIS 6 (6 hours)
7	Basic binary trees	[2-4]	L7 (1 hour)	PW 7 (1 hour)	LW 7 (1 hour)	TSIS 7 (1 hour)	SIS 7 (6 hours)
8	Binary search tree problems	[1-2]	L8 (1 hour)	PW 8 (1 hour)	Mid-term	TSIS 8 (1 hour)	SIS 8 (6 hours)
9	Set, HashSet, LinkedHashSet	[2-5]	L9 (1 hour)	PW 9 (1 hour)	LW 8 (2 hour)	TSIS 9 (1 hour)	SIS 9 (6 hours)
10	Sorted and Tree sets	[5]	L10 (1 hour)	PW 10 (1 hour)	LW 9 (1 hour)	TSIS 10 (1 hour)	SIS 10 (6 hours)
11	Hash table	[2-4]	L11 (1 hour)	PW 11 (1 hour)	LW 10 (1 hour)	TSIS 11 (1 hour)	SIS 11 (6 hours)
12	Advanced hash table	[1-2]	L12 (1 hour)	PW 12 (1 hour)	LW 11 (1 hour)	TSIS 12 (1 hour)	SIS 12 (6 hours)
13	Basic Graphs	[1-4]	L13 (1 hour)	PW 13 (1 hour)	LW 12 (1 hour)	TSIS 13 (1 hour)	SIS 13 (6 hours)
14	Depth and breadth first traversals in graphs	[1-3]	L14 (1 hour)	PW 14 (1 hour)	LW 13 (1 hour)	TSIS 14 (1 hour)	SIS 14 (6 hours)
15	String algorithms	[1]	L15 (1 hour)	PW 15 (1 hour)	Endterm	TSIS 15 (1 hour)	SIS 15 (6 hours)
	Total hours: 150		**15**	**15**	**15**	**15**	**90**

7. List of topics/assignments for laboratory classes

№	Topic Title	Number of hours	References	Form of reporting	Deadline
1	Basic sorting algorithms - Implementation bubble, insertion, selection and merge sorts Implementation of advanced versions of sorting algorithms	2 h	[3-4]	Code	2 week
2	Advanced sorting algorithms Bucket, radix, heap, quick and shell sort implementation	1 h	[1-3]	Code	3 week
3	Generics Implementing previous algorithms using generics to sort objects	1 h	[3-4]	Code	4 week
4	Array, linked lists Implementing add, clone, remove and iterator methods	1 h	[2-4]	Code	5 week
5	Queues and priority queues Implementing add, clone, remove and iterator methods	1 h	[4]	Code	6 week
6	Basic binary trees Implementing add, clone, remove and iterator methods	1 h	[2-4]	Code	7 week
7	Binary search tree problems Implementing add, clone, remove and iterator methods	1 h	[1-2]	Code	8 week
8	Set, HashSet, LinkedHashSet Implementing add, clone, remove and iterator methods	2 h	[2-5]	Code	9 week
9	Sorted and Tree sets Implementing add, clone, remove and iterator methods	1 h	[5]	Code	10 week
10	Hash table Implementing add, clone, remove and iterator methods	1 h	[2-4]	Code	11 week
11	Advanced hash table Implementing add, clone, remove and iterator methods	1 h	[1-2]	Code	12 week
12	Basic Graphs Implementing add, clone, remove and iterator methods	1 h	[1-4]	Code	13 week
13	Depth and breadth first traversals in graphs Implementing DFT and BFT	1 h	[1-3]	code	14 week

8. List of topics/ assignments for practical classes

№	Topic Title	Number of hours	References	Form of reporting	Deadline
1	Big-O notation, recursion Big o notation for sorting and searching algoritms	1 h	[1-2]	code	1 week
2	Basic sorting algorithms - Implementation bubble, insertion, selection and Implementation of advanced versions of sorting algorithms	1 h	[3-4]	Code	2 week
3	Advanced sorting algorithms Bucket, radix, heap, quick and shell sort implementation	1 h	[1-3]	Code	3 week
4	Generics Implementing previous algorithms using generics to sort objects	1 h	[3-4]	Code	4 week
5	Array, linked lists Implementing add, clone, remove and iterator methods	1 h	[2-4]	Code	5 week
6	Queues and priority queues Implementing add, clone, remove and iterator methods	1 h	[4]	Code	6 week
7	Basic binary trees Implementing add, clone, remove and iterator methods	1 h	[2-4]	Code	7 week
8	Binary search tree problems Implementing add, clone, remove and iterator methods	1 h	[1-2]	Code	8 week
9	Set, HashSet, LinkedHashSet Implementing add, clone, remove and iterator methods	1 h	[2-5]	Code	9 week
10	Sorted and Tree sets Implementing add, clone, remove and iterator methods	1 h	[5]	Code	10 week
11	Hash table Implementing add, clone, remove and iterator methods	1 h	[2-4]	Code	11 week
12	Advanced hash table Implementing add, clone, remove and iterator methods	1 h	[1-2]	Code	12 week
13	Basic Graphs Implementing add, clone, remove and iterator methods	1 h	[1-4]	Code	13 week
14	Depth and breadth first traversals in graphs Implementing DFT and BFT	1 h	[1-3]	code	14 week
15	String algorithms Implementing string algorithms	1 h	[1]	code	15 week

9. List of topics/assignments for Student Independent Study

Proper organization of students independent study is the key to the formation of skills in mastering, learning, assimilation and systematization of acquired knowledge, ensuring a high level of academic performance in the learning process

№	Topic Title	Number of hours	References	Form of reporting	Deadline
1	Big-O notation, recursion Big o notation for sorting and searching algoritms	6 h	[1-2]	code	1 week
2	Basic sorting algorithms - Implementation bubble, insertion, selection and merge sorts Implementation of advanced versions of sorting algorithms	6 h	[3-4]	Code	2 week
3	Advanced sorting algorithms Bucket, radix, heap, quick and shell sort implementation	6 h	[1-3]	Code	3 week
4	Generics Implementing previous algorithms using generics to sort objects	6 h	[3-4]	Code	4 week
5	Array, linked lists Implementing add, clone, remove and iterator methods	6 h	[2-4]	Code	5 week
6	Queues and priority queues Implementing add, clone, remove and iterator methods	6 h	[4]	Code	6 week
7	Basic binary trees Implementing add, clone, remove and iterator methods	6 h	[2-4]	Code	7 week
8	Binary search tree problems Implementing add, clone, remove and iterator methods	6 h	[1-2]	Code	8 week
9	Set, HashSet, LinkedHashSet Implementing add, clone, remove and iterator methods	6 h	[2-5]	Code	9 week
10	Sorted and Tree sets Implementing add, clone, remove and iterator methods	6 h	[5]	Code	10 week
11	Hash table Implementing add, clone, remove and iterator methods	6 h	[2-4]	Code	11 week
12	Advanced hash table Implementing add, clone, remove and iterator methods	6 h	[1-2]	Code	12 week
13	Basic Graphs Implementing add, clone, remove and iterator methods	6 h	[1-4]	Code	13 week

14	Depth and breadth first traversals in graphs Implementing DFT and BFT	6 h	[1-3]	code	14 week
15	String algorithms Implementing string algorithms	6 h	[1]	code	15 week

10. System for evaluating student performance in a discipline:

Period	Assignments	Score	Total
1st attestation	**Laboratory works:**	**35**	**100**
	Lab work 1,	5	
	Lab work 2,	5	
	Lab work 3,	5	
	Lab work 4,	5	
	Lab work 5,	5	
	Lab work 6,	5	
	Lab work 7,	5	
	Practical lessons:	**21**	
	Exercise 1,	3	
	Exercise 2,	3	
	Exercise 3,	3	
	Exercise 4,	3	
	Exercise 5,	3	
	Exercise 6,	3	
	Exercise 7,	3	
	Mid-term	**30**	
	SIS assignments	**14**	
2nd attestation	**Laboratory works:**	**36**	**100**
	Lab work 8,	6	
	Lab work 9,	6	
	Lab work 10,	6	
	Lab work 11,	6	
	Lab work 12,	6	
	Lab work 13,	6	
	Practical lessons:	**24**	
	Exercise 8,	3	
	Exercise 9,	3	
	Exercise 10,	3	
	Exercise 11,	3	
	Exercise 12,	3	
	Exercise 13,	3	
	Exercise 14,	3	
	Exercise 15,	3	
	End-of-term	**30**	
	SIS assignments	**10**	
Exam			**100**
Total	**0,3*1stAtt+0,3*2ndAtt+0,4*Final**		

11. Assessment criteria:

Letter Grade	Numerical equivalent	Points (%)	Traditional system assessment	General description of grading criteria
A	4,0	95-100	Excellent	The student has knowledge of the subject in the full scope of the curriculum, understands the discipline deeply enough; shows a high level of knowledge that exceeds the volume provided by the syllabus, gives an exhaustive answer
A-	3,67	90-94		The student has knowledge of the subject in the full scope of the curriculum, understands the discipline deeply enough; gives an exhaustive answer
B+	3,33	85-89	Good	The student shows a complete, well-founded knowledge of the subject, but the answers did not always highlight the main idea, rational methods of calculation were not always used; the answers were mostly brief and sometimes unclear.
B	3,0	80-84		
B-	2,67	75-79		
C+	2,33	70-74		
C	2,0	65-69	Satisfactory	The student demonstrates sufficient knowledge of the subject, but without proper depth and justification, the answers are unclear and without proper logical sequence.
C-	1,67	60-64		
D+	1,33	55-59		
D	1,0	50-54		
FX	0,5	25-49	Unsatisfactory	The student demonstrates insufficient knowledge of the subject, positive answers were not given to individual questions.
F	0	0-24		The student demonstrates a very low level of knowledge of the subject.