جامعة فيلادلفيا Philadelphia University	اسم النموذج : خطة المادة الدراسية	رمز النموذج : QFO-AP-VA-008
	الجهة المصدرة: نائب الرئيس للشؤون الأكاديمية	رقم الإصدار: (Rev) 2
	الجهة المدققة : اللجنة العليا لضمان الجودة	تاريخ الإصدار: 20/09/2023
		عدد صفحات النموذج: 6

Course Title: Theory of Computation			**Course code:** 0750224		
Course Level: Bachelor			**Course prerequisite (s) and/or corequisite(s):** 0750120		
Lecture Time:			**Credit hours:** 3		
UR ☐	**FR** ☐	**DR** ☐	**C** ☐	**E** ☐	

Academic Staff Specifics

Name	Rank	Office No. and Location	Office Hours	E-mail Address
Dr. AL-Oudat Mohammad	Assistant Professor	IT Building 2nd Floor	TBA	maloudat@philadelphia.edu.jo

The Learning Style Used in Teaching the Course

The Learning Style			
Blended Learning ☐			
Electronic Learning ☐			
Face-to-Face Learning ☑			
Face-to-Face	**Electronic**	**Blended**	**Percentage**
💯			

Course/Module Description

The "Theory of Computation" course provides students with a solid foundation in the fundamental concepts of theoretical computer science. Throughout this 16-week journey, students will delve into the mathematical and conceptual underpinnings of computation. Topics covered include mathematical preliminaries and notation, the three basic concepts of languages, grammar, and automata, as well as regular sets and expressions.

Students will gain a deep understanding of finite automata, including deterministic and non-deterministic finite automata, and explore their equivalence. They will master regular expressions and grammar and learn how to manipulate them. The course delves into context-

free grammar, parsing techniques, and methods for transforming grammar into specific normal forms.

In the latter part of the course, students will explore advanced topics, including pushdown automata and their connection to context-free languages. The course culminates with an introduction to Turing machines, the theoretical foundation of computation, and discussions on Turing's Thesis.

Throughout the course, students will engage in hands-on exercises and examples to solidify their understanding of these theoretical concepts. By the end of this course, students will possess the knowledge and analytical skills necessary to comprehend the theoretical framework that underlies modern computing, setting a strong foundation for further studies in computer science and related fields.

Course/Module Objectives:

At the end of the course, students will be able to:

- Foundational Knowledge: Develop a strong foundation in theoretical computer science, including formal languages, grammars, and automata.
- Language and Grammar Mastery: Master regular languages, regular expressions, context-free grammars, and parsing techniques.
- Problem-Solving Skills: Enhance problem-solving abilities by applying theoretical concepts to practical examples and exercises.
- Theoretical Computation Analysis: Gain the ability to analyze and compare the computational power of various automata models and formal grammars.
- Preparation for Advanced Studies: Build a solid groundwork for advanced studies in theoretical computer science and related fields, providing a strong academic foundation.

Course/ module components

- **Textbook:**
  - M. Sipser, "Introduction to the Theory of Computation," Cengage Learning, 2012.
- **Supporting material(s):**
  - J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Automata Theory, Languages, and Computation," Addison-Wesley, 2006.
  - D. C. Kozen, "Theory of Computation," Springer, 2006.
  - S. Arora and B. Barak, "Computational Complexity: A Modern Approach," Cambridge University Press, 2009.
  - P. Linz, "Introduction to Formal Languages and Automata," Jones & Bartlett Learning, 2012.

Teaching methods:

- A1, A2, B1, and D1 are achieved through lectures and assessed by quizzes and examinations.

- B2, and C1 are achieved and assessed by homework: practical and research works.

Learning outcomes

Knowledge:

- A1: Deep Understanding of theoretical computer science, including formal languages, grammars, automata, and Turing machines.
- A2: Research Methodologies for analyzing computational problems and creating formal models.

Objective B: Skills

- B1: Critical Thinking skills to devise original solutions within automata, grammars, and formal languages.
- B2: Problem-Solving Proficiency for addressing complex theoretical challenges.

Objective C: Skills (continued)

- C1: Applied Knowledge in areas like regular expressions, automata, and parsing techniques.

Objective D: Competencies

- D1: Effective Communication skills for conveying complex theoretical concepts.

Learning outcomes achievement

Development:

- Assessment:
- **Assessment instruments**

Allocation of Marks	
Assessment Instruments	**Mark**
mid examination	**30%**
Final examination	**40%**
Quizzes, and tutorial contributions	**30%**
Total	**100%**

Documentation and academic honesty

Submit your homework covered with a sheet containing your name, number, course title and number, and type and number of the homework (e.g. tutorial, assignment, and project).

Any completed homework must be handed in to my office by 15:00 on the due date. After the deadline "zero" will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

For the research report, you are required to write a report similar to a research paper. It should include:

- **Abstract**: It describes the main synopsis of your paper.
- **Introduction:** It provides background information necessary to understand the research and get readers interested in your subject. The introduction is where you put your problem in

context and is likely where the bulk of your sources will appear.

- **Methods (Algorithms and Implementation)**: Describe your methods here. Summarize the algorithms generally, highlight features relevant to your project, and refer readers to your references for further details.

- **Results and Discussion (Benchmarking and Analysis)**: This section is the most important part of your paper. It is here that you demonstrate the work you have accomplished on this project and explain its significance. The quality of your analysis will impact your final grade more than any other component on the paper. You should therefore plan to spend the bulk of your project time not just gathering data, but determining what it ultimately means and deciding how best to showcase these findings.

- **Conclusion**: The conclusion should give your reader the points to "take home" from your paper. It should state clearly what your results demonstrate about the problem you were tackling in the paper. It should also generalize your findings, putting them into a useful context that can be built upon. All generalizations should be supported by your data, however; the discussion should prove these points, so that when the reader gets to the conclusion, the statements are logical and seem self-evident.

- **Bibliography:** Refer to any reference that you used in your assignment. Citations in the body of the paper should refer to a bibliography at the end of the paper.

- **Protection by Copyright**

1. **Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.**
2. Use of quotations or data from the work of others is entirely acceptable and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

- **Avoiding Plagiarism**

1. **Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.**

2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a

few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.

3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.

4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty, the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/Module Academic Calendar

Week	Basic and support material to be covered	Homework/reports and their due dates	Lab works and tutorials
1	Mathematical Preliminaries and Notation		Lab 01
2	Three Basic Concepts: Languages, Grammars, Automata		Lab 02
3	Introduction to Regular Expressions		Lab 03
4	Deterministic Finite Automata (DFA) - Part 1		Lab 04
5	DFA - Part 2 (Examples)		Lab 05
6	Non-deterministic Finite Automata (NFA)	***First Assignment***	Lab 06
7	NFA with Lambda-Transitions, DFA to NFA		Lab 07
8	***Mid Term Exam Period***		
9	NFA to DFA, Finite State Acceptance of Regular Languages		Lab 08
10	Equivalence of DFA and NFA, Regular Expressions (RE)		Lab 09
11	RE Examples, DFA to RE, Handling Non-Regular Languages		Lab 10
12	Context-Free Grammar (CFG) - Part 1 (Leftmost Derivation and Ambiguity)		Lab 11
13	Parsing	***Second Assignment/Project***	Lab 12
14	Two Important Normal Forms (Chomsky Normal Form)		Lab 13
15	Two Important Normal Forms (Greibach Normal Form)		Lab 14
16	***Final Exam Period***		

Expected workload:

On average students need to spend 3 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy:

- Absence from lectures and/or tutorials shall not exceed 15%.
- Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course.
- If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module references

- N/A

Website(s):

1. **MIT OpenCourseWare - Introduction to Theory of Computation: MIT's course materials, including lecture notes and video lectures, offer a comprehensive overview of theory of computation concepts.**
    - **Website: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2020/index.htm**
2. **GeeksforGeeks - Theory of Computation: GeeksforGeeks provides free articles, tutorials, and practice problems for in-depth learning and problem-solving in automata theory and formal languages.**
    - **Website: https://www.geeksforgeeks.org/theory-of-computation-2/**
3. **Brilliant - Theory of Computation Course: Brilliant offers an interactive, free course that covers theory of computation topics, providing a hands-on learning experience.**
    - **Website: https://brilliant.org/courses/theory-of-computation/**
4. **TOC@Wiki: TOC@Wiki is a community-driven wiki that covers theoretical computer science topics, making it a valuable reference for understanding fundamental concepts.**
    - **Website: https://toc.wiki/**
5. **YouTube: YouTube hosts various educational channels and playlists related to theory of computation. Channels like "Computerphile" and "Neso Academy" offer video lectures and tutorials.**
    - **YouTube Channels:**
        - **Computerphile: https://www.youtube.com/user/Computerphile**
        - **Neso Academy: https://www.youtube.com/user/nesoacademy**

These five resources provide a solid foundation for your studies in the theory of computation and can assist you in understanding core concepts, practicing problem-solving, and gaining insights into this field.

-