KENYATTA UNIVERSITY
SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTING AND INFORMATION TECHNOLOGY
SIT 211: INTRODUCTION TO LOGIC PROGRAMMING
COURSE OUTLINE

Lecturer: *Gordon Agutu,*
Office: *Engineering Building, Room 217*
Email**:** *gagutu@gmail.com* ***Tel:*** *+254 733 712 991 or +254 727 342 682*
Class Hours: *Wednesday 11h00 – 14h00 (Computer Lab)* **Consultation Hours:** *Wednesday 14h00 – 19h00*

Aim:
To introduce students to Logic programming and provide students with an understanding of Prolog programming language

Course Learning Outcomes:
After successfully completing this course, the student should be able to:

- Explain the differences between the declarative and procedural programming paradigms
- Describe the syntax and semantics of propositional and predicate logic
- Develop and illustrate the execution of non-trivial Prolog programs
- Discuss the potential applications of Prolog programming language
- Apply Prolog's built-in predicates in obtaining input and producing output
- Identify the strengths and weaknesses of Prolog

Contact Hours: 35

Prerequisite: SIT 102: Introduction to Structured Programming
SIT 108: Introduction to Logic

Indicative Content
Introduction to Logic Programming Environment. A closer look at Prolog: Facts, Facts with arguments, rules, computation. Variables and Unifications: Simple unification Operators and Structures Back Tracking: Fail, Cut, Not Lists and Strings: List manipulation Built-in Predicates: Input, Output, Control, Data Manipulation, Arithmetic Meta-logic and I/O Parsing: Low-level and high-level parsing expressions, Parsing optional structures. Prolog semantics: Computational model, Unification, resolution, queries

Content

Week/Dates	Topic	Intended Learning Outcomes	Activities
Week 1	**Introduction** ▪ Course Outline ▪ Recap - Structured Programming and Introduction to Logic	▪ Understand what the course entails ▪ Demonstrate an understanding of concepts covered in the prerequisite courses.	▪ Introduction ▪ Recap Questions and Answers
Week 2	▪ Introduction to Logic Programming Environment	▪ Explain the need for logic programming ▪ Distinguish between functional programming and logic programming ▪ Briefly describe propositional logic ▪ Briefly describe predicate logic	▪ Class discussions ▪ Class exercises ▪ Group discussions
Week 3	▪ A closer look at Prolog ▪ Facts, Facts with arguments ▪ Rules ▪ Computation rules	▪ Discuss the potential applications of Prolog programming language ▪ Identify the strengths and weaknesses of Prolog ▪ Define facts and rules ▪ Illustrate application of facts and rules in computations	▪ Class discussions ▪ Class exercises ▪ Lab Exercises
Week 4	▪ Variables and Unifications ▪ Simple unification Operators and Structures	▪ Describe variables and explain their applications in Prolog ▪ Describe the various features of variables	▪ Class discussions ▪ Lab Exercises ▪ Group Exercises

		▪ Describe operator notation for predicates	
Week 5	▪ Back Tracking ▪ Fail, repeat, and Not ▪ Cut and Cut with failure ▪ CAT 1:	▪ Demonstrate backtracking ▪ Employ Prolog's control facilities such as Cut, Not, and repeat ▪ Identify green and red cuts ▪ Explain the difference between green and red cuts ▪ Analyse the issues associated with negation in goals	▪ Research assignments ▪ Class exercises ▪ Lab Exercises CAT 1
Week 6	▪ Lists and Strings ▪ List manipulation ▪ Built-in Predicates	▪ Create a list containing all the possible values that would satisfy a specified goal ▪ Illustrate the execution of predicates that manipulate lists and strings ▪ Analyze the execution of built-in and recursive predicates ▪ Demonstrate how to work through a list element by element	▪ Lab Exercises ▪ Class exercises ▪ Tutorials
Week 7	▪ Input and Output ▪ Control ▪ Data Manipulation	▪ Describe the built-in predicates for input and output operations ▪ Use Prolog's built-in predicates for obtaining input and producing output	▪ Lab Exercises ▪ Class exercises ▪ Tutorials
Week 8	▪ Arithmetic expressions ▪ Meta-logic and I/O Parsing ▪ Low-level and high-level parsing expressions	▪ Explain the execution of predicates that manipulate arithmetic expressions and structures ▪ Describe the operators used for evaluating and comparing the values of arithmetic expressions ▪ Define and test a predicates that take various arguments	▪ Lab Exercises ▪ Group exercises ▪ Class discussions
Week 9	▪ Parsing optional structures ▪ Prolog semantics ▪ Computational model ▪ CAT 2	▪ Describe semantics of prolog ▪ Explain how parsing is carried out for optional structures ▪ Develop a computational model	▪ Lab Exercises ▪ Class exercises ▪ Group discussions ▪ CAT 2
Week 10	▪ Unification ▪ Resolution ▪ Queries	▪ Illustrate application of unification in prolog ▪ Demonstrate the use of queries in prolog	▪ Lab Exercises ▪ Class discussions ▪ Class exercises ▪ Group discussions
Week 11	▪ Summary and Tutorial	▪ Recall and apply major concepts covered	▪ Tutorial
Week 12	▪ Preparation for final exams	▪ Attempt questions of various areas of the course content	▪ Consultations ▪ Revision for exams

Course Delivery Methodology

- Lectures: At least 2 hours per week of lectures will be used to introduce material on the formal aspects of the unit
- Tutorials: 1 hour per week
- Laboratory Exercises: At least three sessions per semester each being 3 hours long
- Students will research and present their findings on various topics
- Discussions and working out problems

Academic Assessment

- Examination: 70%
- CATs: 15%, Assignments and lab exercises: 15%
- Computer Laboratory Exercises: Each student is expected to carry out at least seven computer laboratory exercises. These will include lab exercises done using Prolog

Course Reference Materials

- Michael Spivey, An introduction to logic programming through Prolog, c Prentice–Hall International, 2008
- Max Bramer, Logic Programming with Prolog, Springer Science+Business Media, 2005 ISBN-10: 1-85233-938-1, ISBN-13: 978-1852-33938-8
- Ivan Bratko, Prolog-Programming for Artificial Intelligence, 3rd Edition, Addison Wesley, 2001.
- Leon Sterling and Ehud Shapiro, The Art of Prolog, 2nd Edition, MIT Press, 1994.
- W. F. Clocksin, C. S. Mellish, Programming in Prolog: Using the Iso Standard, 5th edition, Springer-Verlag, 2003.

Classes

- Punctuality is fundamental.
- Active participation in class discussions is encouraged

Assignments and/or Course Work

- Plagiarism is a serious offence. If detected in any form in course work and assignments, the following will apply:
    - In partial or non-serious cases (such as not citing whole word-for-word quotes), half the total possible marks of the assignment are duly struck off.
    - In serious cases (such as whole duplication of a paper), a zero policy will apply i.e., all offending assignments will be awarded a mark of zero.
    - *Note:* The level of seriousness referred to above is at the discretion of the lecturer. Appeals are certainly possible through the relevant channels
- Notwithstanding the above, collaboration in course work is certainly encouraged as this promotes team spirit and group synergy as long provided originality is preserved.

Communication Channels

- E-mail: To make appointments or to inform the lecturer of any absenteeism from class prior to the class. However you may later be required to meet the lecturer to explain the absence
- Class representative