**1. All the lecture notes (written in English) are free available at following G-Drive link;**
https://drive.google.com/drive/folders/1qw7JdVLxD-8AXOaWEytD3kbng8a6znaU

**2. All the lecture notes (with License - free to reuse) are available at GitHub link;**
https://github.com/DrUmairAbdullah/Introduction-to-Artificial-Intelligence-using-Python/releases/tag/v1.0

**3. All the Video lectures (in Urdu) are freely available at the following playlist.**
https://www.youtube.com/playlist?list=PLvzUfZlNIColrIsWvdNKQyKCI6tFWN7N4

# Artificial Intelligence Course Overview

**Instructor:** Umair Abdullah
**Course Format:** Weekly YouTube Lectures + Python Code Examples
**Language:** Urdu (Lectures) & English (Notes)
**Level:** Undergraduate (Beginner to Intermediate)

---

## Course Abstract

This course offers a comprehensive introduction to **Artificial Intelligence (AI)**, combining conceptual foundations with hands-on programming experience. Delivered in **Urdu** with accompanying **English lecture notes**, the course targets undergraduate students, educators, and self-learners in computer science and related fields. Over 16 weeks, learners will explore essential AI topics such as knowledge representation, heuristic search, production systems, classical expert systems, case studies (GPS, ELIZA, STUDENT), and logic programming using Prolog. In addition to theoretical insights, the course integrates Python programming from basic constructs to advanced recursive techniques applied to AI problems. Through structured case studies and coding walkthroughs, students develop practical skills relevant to real-world AI applications. This dual emphasis on theory and implementation provides a strong foundation for further study or entry-level work in artificial intelligence and software development.

## Topics not included in this course

This introductory course on Artificial Intelligence is designed to provide foundational knowledge in symbolic reasoning, expert systems, classical search strategies, production systems, and logic programming using Prolog. However, several advanced or specialized areas within the broader AI domain are intentionally excluded from this course. These include robotics, speech and voice processing, computer vision, and reasoning under uncertainty such as abduction, fuzzy logic, and probabilistic models like Bayesian networks.

Modern areas of AI such as machine learning**,** neural networks, and deep learning are not covered in this course. Correspondingly, learners will not engage with popular Python libraries such as Scikit-learn**,** TensorFlow, or PyTorch, which are typically used in training and deploying data-driven AI models. Topics such as transformer architectures, Large Language Models (LLMs) like GPT**,** BERT**,** and their implementations using libraries like Hugging Face Transformers are also beyond the scope of this module. Additionally, the course does not address topics in evolutionary computation (e.g., genetic algorithms), production rule shells, memory-based learning systems, or other advanced areas such as meta-knowledge, intelligent databases, and agent-based AI frameworks. These topics are recommended for learners pursuing advanced AI studies, particularly in applied machine learning, data science, or neural computation, following completion of this foundational course.

## Weekly Course Contents

| Week | Topics | YouTube link |
|---|---|---|
| 1 | **Introduction to AI:** Course learning objectives are outlined, followed by a formal definition of AI. The lecture introduces the major domains of AI and provides an overview of how these areas contribute to the development of AI systems. | https://youtu.be/uOd-ku_7xsw<br>https://youtu.be/PTzopLwLpgQ<br>https://youtu.be/mPKA9zeKFE4<br>https://youtu.be/vRQvXWbdO5w |
| 2 | **The Dream:** Explores the visionary goals of AI and the evolving research areas such as planning, perception, and natural language understanding. Introduces expert systems, their architecture, and classical examples including prototypes and industry applications, highlighting how domain-specific knowledge enables intelligent decision-making. | https://youtu.be/nd59HQMId4A<br>https://youtu.be/H2XdcfFr_-8<br>https://youtu.be/Ot6mOX5FOEU<br>https://youtu.be/g4lJB_VBX4E<br>https://youtu.be/5FhGkd5CRy4<br>https://youtu.be/G5Sg4MLtiLw<br>https://youtu.be/f3Pr1fwxAe4 |
| 3 | **Python for AI – Basics:** Covers the essential features needed in AI development tools, followed by a beginner-friendly introduction to Python. Key programming concepts include expression evaluation, function calls, and basic list operations such as creation, access, and manipulation — forming the foundation for implementing AI logic. | https://youtu.be/Q0B49yZpmic<br>https://youtu.be/_7Lymc3wSYE<br>https://youtu.be/n8W9BG7OQCw<br>https://youtu.be/_s7UX4Jj9MM<br>https://youtu.be/zBAA9zNjG6M<br>https://youtu.be/YIIgm1J5hvY |
| 4 | **Knowledge Representation:** Introduces the concept and significance of knowledge representation in AI systems, along with its key requirements. Discusses various methods such as logic-based representations, production rules, semantic networks, frames etc. | https://youtu.be/Dy4Tt9TEO00<br>https://youtu.be/g8Jc5mIVmXE<br>https://youtu.be/Rf03InHOmbA<br>https://youtu.be/dPsxB-IfkGM<br>https://youtu.be/xR3Mk-RnMYE |
| 5 | **Production Systems & Programming Examples:** Covers the structure and working of production systems with a focus on OPS-5. Demonstrates programming with rule-based systems using practical examples such as calculating the area of a circle, factorials etc. | https://youtu.be/yhvYrTbMMuc<br>https://youtu.be/d-_ZEIEeJdM<br>https://youtu.be/NPrwSLeCIZg<br>https://youtu.be/XBNN8H61azI<br>https://youtu.be/3Vr6W-q_C1Q |
| 6 | **Search Techniques – Part 1:** Introduces search as a fundamental AI problem-solving strategy. Covers uninformed and heuristic-based methods including depth-first search (DFS), breadth-first search (BFS), hill climbing, beam search, and best-first search, highlighting their use in navigating large solution spaces. | https://youtu.be/WCIlRnkntFI<br>https://youtu.be/mYG5ueX3Pyo<br>https://youtu.be/dRO8e3Dc4pY<br>https://youtu.be/7oAAEm8xJt4<br>https://youtu.be/CuLhji1wpAQ<br>https://youtu.be/y1_VISG0Nlk<br>https://youtu.be/KO1g7Zl-ftU<br>https://youtu.be/cPBTOPp8IVg |
| 7 | **Search Techniques – Part 2:** This week delves into advanced search techniques used for identifying optimal solutions within complex problem spaces. Discussing classical algorithms like the British Museum approach, which attempts exhaustive search, and then move to more practical techniques such as Branch and Bound, along with its refined variants — including underestimation and integration with dynamic programming to prune inefficient paths. The widely-used A* search algorithm is discussed in depth, highlighting its ability to balance cost estimation with optimality. | https://youtu.be/_YK8Xrk8uN0<br>https://youtu.be/WOBEZHOk7TU<br>https://youtu.be/lBpRIlCuV5k<br>https://youtu.be/D8sSOa-rQjs<br>https://youtu.be/9jWSxRZpZEw<br>https://youtu.be/uL2luAbChQg<br>https://youtu.be/xA-6jzpT2K8<br>https://youtu.be/QkXIGNnlSYU<br>https://youtu.be/T-RMRkHUvTE<br>https://youtu.be/a8v-0GPfaJA<br>https://youtu.be/xtE_QZ7KIyg<br>https://youtu.be/TX-O9cdYpY4 |
| 8 | **Iterative & Recursive Programming in Python:** This week introduces two fundamental programming paradigms—iteration and recursion—which are essential for implementing algorithms in AI and beyond. Students begin with iterative programming constructs such as for and while loops, gaining hands-on practice with repeated operations and control flow. The session then transitions into recursive programming, where problems are solved by defining functions that call themselves with modified parameters. Emphasis is placed on understanding recursion depth, base | https://youtu.be/y8TxCmqOrcY<br>https://youtu.be/GjdGjmzwSNA<br>https://youtu.be/v9z6os2M71o<br>https://youtu.be/IxLo7pXJKN8<br>https://youtu.be/prxtWuv4snY<br>https://youtu.be/UHQD_tzH4y4<br>https://youtu.be/e-_9B6wjxJo<br>https://youtu.be/EBoSUpEswjk<br>https://youtu.be/YpbBk7esFmU<br>https://youtu.be/nUj9XdIqf1I |

| | | |
|---|---|---|
| | cases, and how recursion can be applied elegantly to solve problems involving nested structures or divide-and-conquer strategies. Examples are provided to contrast iterative and recursive solutions for the same tasks, helping students develop an intuition for selecting the right approach. These skills are foundational for later AI topics such as tree traversal, search algorithms, and symbolic processing. | https://youtu.be/cC2ccGZ8Yyk<br>https://youtu.be/LlvPoAhcttc<br>https://youtu.be/pckK0KwBsqo<br>https://youtu.be/bk7xdVwBYXU<br>https://youtu.be/mlB-ebdtMQY<br>https://youtu.be/zmsbkJqI-TA<br>https://youtu.be/_684F8wXUPw |
| 9 | **Advance recursive programming in Python:** Covers deeper aspects of recursion, including passing additional parameters to recursive functions for greater control and flexibility. Demonstrates recursive techniques for processing complex and nested data structures such as lists within lists and tree-like formats. | https://youtu.be/MBecHBhkU8o<br>https://youtu.be/zhWgWD28_Ew<br>https://youtu.be/gWs8ooNSLYY<br>https://youtu.be/fibHOcU-Tcw<br>https://youtu.be/uAVgwPQCRZw<br>https://youtu.be/Y7tKlcfUE00<br>https://youtu.be/dNc-fleLgac |
| 10 | **Case Study 1 - General Problem Solver:** Introduces the General Problem Solver as a classical AI planning system designed to model human-like problem solving. Includes discussion of its architecture and working mechanism with examples. Code understanding is optionally covered to illustrate how GPS applies means-ends analysis to reach goals. The session highlights how declarative knowledge and goal decomposition enable GPS to solve a wide range of symbolic problems, laying the groundwork for modern AI planning techniques. | https://youtu.be/8TqfwxEOHtA<br>https://youtu.be/PJmhdvr2TsE<br>https://youtu.be/EkJUHSzI_Aw<br>https://youtu.be/cTjo9jSToJQ<br>https://youtu.be/1ljVgtDEMwo<br>https://youtu.be/ran8kLIPWzM<br>https://youtu.be/tgM9BU6PnCQ<br>https://youtu.be/9FGxOsWXmhE<br>https://youtu.be/92eyE4Mf9Uc<br>https://youtu.be/DCqiGghEkDw |
| 11 | **Case Study 2 - ELIZA:** Introduces ELIZA, an early natural language processing program that simulates conversation using pattern matching. Discusses its relevance to the Turing Test and explores the underlying logic. Includes a walkthrough of Python code that mimics ELIZA's pattern matching behavior. | https://youtu.be/M6dYNaj3HkU<br>https://youtu.be/ExBXxrXxUI0<br>https://youtu.be/1r8jphsVAAM<br>https://youtu.be/aXHcBQqkLnk<br>https://youtu.be/vreqK3xkzXU |
| 12 | **Segment Matching in Python:** Covers the segment matching module used in natural language processing and symbolic reasoning. Focuses on tracing and understanding the Python code implementation, with emphasis on how patterns and variable bindings are matched in structured input. | https://youtu.be/aMThSK5MStU<br>https://youtu.be/3CzM0ECVFgY<br>https://youtu.be/Da35ZA-Hc-E<br>https://youtu.be/fm2p3B02Zis<br>https://youtu.be/xWMMNTwfljo |
| 13 | **Case Study 3 - STUDENT program in Python:** Introduces the STUDENT program, a classic AI system for solving algebraic word problems. Focuses on the "translate to expression" module, with detailed tracing of Python code to understand how natural language input is converted into structured mathematical expressions. | https://youtu.be/6fLbQFA3EgQ<br>https://youtu.be/U8cIFnIhGzM<br>https://youtu.be/luk97qZePxw<br>https://youtu.be/s7Zh-5HJCp4<br>https://youtu.be/mBPj61n9jSw |
| 14 | **Case Study – STUDENT Program (Equation Solving):** Focuses on the equation-solving module of the STUDENT program. Explains how the system identifies equations with one variable, isolates the variable, and solves it step by step. Python code is examined to trace the logic and symbolic manipulation involved in reaching a solution. | https://youtu.be/488tG5BOML8<br>https://youtu.be/TL2q_CldUr4<br>https://youtu.be/OT6Ay8b4Ifc<br>https://youtu.be/fSw-nH7Lomw<br>https://youtu.be/KKB7ZUYDrWc<br>https://youtu.be/pWSvQue4rlQ |
| 15 | **Logic and Inference:** Introduces the fundamentals of logic in the context of AI, including propositional and predicate logic. Explains the role of deductive reasoning in drawing valid conclusions from known facts. The session focuses on applying inference rules—such as modus ponens and resolution—to prove the validity of logical arguments, forming the basis for reasoning systems in AI. | https://youtu.be/BqQWYL3H0uw<br>https://youtu.be/XRbw8wk3uqo<br>https://youtu.be/BqQWYL3H0uw<br>https://youtu.be/pEkjHe48KC8<br>https://youtu.be/B59BEd7grgc<br>https://youtu.be/XA5RSPqXs6Y<br>https://youtu.be/As42vTbng-c<br>https://youtu.be/xbJ_001Zn_0 |
| 16 | **Introduction to Prolog:** What is Prolog, Example of relationships (Family Tree Example) in Prolog.<br>Defining new relations, using facts and existing relations.<br>Defining and executing simple programs in prolog. | https://youtu.be/xbLATtBL01o<br>https://youtu.be/NhNF32ZYx20<br>https://youtu.be/m6V4qnmHvRo<br>https://youtu.be/J1bCRxEWBaI<br>https://youtu.be/Rkg0gfMBRx4<br>https://youtu.be/4UzBgxpBdxs |

**Reference Books:**

- "Paradigms of Artificial Intelligence" by Peter Norvig
- "Understanding Artificial Intelligence" Sams Understanding Series
- "Artificial Intelligence a Modern Approach" by Russell and Norvig
- "Artificial Intelligence Structures and Strategies for Complex Problem Solving" by Luger and Stubblefield

**Reference Websites:**

- www.ai.mit.edu
- www.aaai.org
- www.ai.sri.com
- www.aiai.ed.ac.uk
- www.a-i.com
- www.pcai.com
- www.cs.berkeley.edu
- www.gameai.com

---

### Simple Steps for Downloading All Lecture Notes (.docx, .pptx) from GitHub

1. Go to the public release (**with License - free to reuse with credit**) page

Link: https://github.com/DrUmairAbdullah/Introduction-to-Artificial-Intelligence-using-Python/releases/tag/v1.0

2. Scroll down to 'Assets' section
3. Click "Introduction-to-Artificial-Intelligence-using-Python-main_v1.0.zip"

### Simple Steps for Downloading All Lecture Notes (.docx, .pptx) from G-Drive

1. Go to G-Drive
   Link: https://drive.google.com/drive/folders/1qw7JdVLxD-8AXOaWEytD3kbng8a6znaU
2. Click the "Download All" button near the top right.

----------END----------