LAPT

LONDON ACADEMY OF PROFESSIONAL TRAINING

CERTIFICATE

Certificate in Python and Artificial Intelligence



LAPT — London Academy of Professional Training

Address 85 Great Portland Street, W1W 7LT, London, United Kingdom

Email info@lapt.org

Telephone +44 7513 283044

Web lapt.org

Reference ITC-C04 • IT Centres

UKPRN 10067351 • Registered in England and Wales, company 4984798

4	68	480	400	60%	Lifetime
SUBJECTS	CHAPTERS	CLASSES	TOTAL MARKS	PASS MARK	VALIDITY

AT A GLANCE

REFERENCE	**ITC-C04**
AWARD	**Certificate**
ASSESSMENT	**400 marks — 240 to pass (60%)**
PREREQUISITES	**None**
VALIDITY	**Lifetime**
AWARDING BODY	**LAPT — London Academy of Professional Training**

CURRICULUM

1 **Python Programming** 12 chapters · 120 classes **100 marks**

- **1 Getting Started with Python and Development Tools** — 10 classes
  - 1.1 What Python Is Used For
  - 1.2 Installing Python on Your Computer
  - 1.3 Using the Interactive Shell
  - 1.4 Choosing and Setting Up a Code Editor
  - 1.5 Writing and Running Your First Script
  - 1.6 Understanding Syntax and Indentation
  - 1.7 Comments and Readable Code
  - 1.8 Reading Error Messages
  - 1.9 Using Virtual Environments
  - 1.10 Organising a Small Project Folder
- **2 Variables, Data Types and Operators** — 10 classes
  - 2.1 Naming and Assigning Variables
  - 2.2 Numbers and Arithmetic
  - 2.3 Text Values and Quotes
  - 2.4 Boolean Values and Comparisons
  - 2.5 Logical Operators
  - 2.6 Type Checking and Conversion
  - 2.7 Getting Input from Users
  - 2.8 Formatting Printed Output
  - 2.9 Operator Precedence
  - 2.10 Practice Building a Bill Calculator
- **3 Strings and Text Processing** — 10 classes
  - 3.1 Indexing and Slicing Strings
  - 3.2 Common String Methods
  - 3.3 Searching Within Text
  - 3.4 Replacing and Splitting Text
  - 3.5 Joining and Cleaning Text
  - 3.6 Formatted String Literals
  - 3.7 Escape Characters and Multiline Text
  - 3.8 Validating Text Input
  - 3.9 Introduction to Regular Expressions
  - 3.10 Practice Cleaning Customer Names
- **4 Decision Making and Loops** — 10 classes
  - 4.1 If and Else Statements
  - 4.2 Multiple Conditions with Elif
  - 4.3 Nested Decisions
  - 4.4 While Loops
  - 4.5 For Loops and Ranges
  - 4.6 Loop Control with Break and Continue
  - 4.7 Looping Over Text
  - 4.8 Common Loop Patterns
  - 4.9 Avoiding Infinite Loops
  - 4.10 Practice Building a Number Guessing Game
- **5 Lists, Tuples, Sets and Dictionaries** — 10 classes
  - 5.1 Creating and Changing Lists
  - 5.2 List Methods and Slicing
  - 5.3 Looping Through Lists
  - 5.4 Tuples and Immutability
  - 5.5 Sets and Removing Duplicates
  - 5.6 Dictionaries and Key Value Pairs
  - 5.7 Nested Data Structures
  - 5.8 List Comprehensions
  - 5.9 Sorting and Searching Collections
  - 5.10 Practice Building a Contact Book
- **6 Functions and Modules** — 10 classes
  - 6.1 Defining and Calling Functions
  - 6.2 Parameters and Arguments
  - 6.3 Return Values
  - 6.4 Default and Keyword Arguments
  - 6.5 Variable Scope
  - 6.6 Writing Helpful Docstrings
  - 6.7 Using Built-In Modules
  - 6.8 Creating Your Own Modules
  - 6.9 Importing Code Across Files
  - 6.10 Practice Refactoring a Script into Functions
- **7 File Handling and Data Formats** — 10 classes
  - 7.1 Opening and Closing Files
  - 7.2 Reading Text Files
  - 7.3 Writing and Appending to Files
  - 7.4 Working with File Paths
  - 7.5 Reading and Writing Delimited Files
  - 7.6 Working with JSON Data
  - 7.7 Handling Large Files Line by Line
  - 7.8 Processing a Sales Data File
  - 7.9 Saving Results to a Report
  - 7.10 Practice Building a Log File Analyser
- **8 Errors, Exceptions and Debugging** — 10 classes
  - 8.1 Types of Programming Errors
  - 8.2 Reading Tracebacks
  - 8.3 Handling Exceptions with Try and Except
  - 8.4 Raising Your Own Exceptions
  - 8.5 Cleaning Up with Finally
  - 8.6 Using Print Statements to Debug
  - 8.7 Using a Debugger with Breakpoints
  - 8.8 Writing Simple Assertions
  - 8.9 Writing Basic Unit Tests
  - 8.10 Practice Fixing a Broken Program
- **9 Object Oriented Programming in Python** — 10 classes
  - 9.1 Thinking in Objects
  - 9.2 Defining Classes and Creating Objects
  - 9.3 Attributes and Methods
  - 9.4 The Constructor Method
  - 9.5 Class and Instance Data
  - 9.6 Encapsulation and Private Data
  - 9.7 Inheritance Between Classes
  - 9.8 Method Overriding
  - 9.9 Special Methods for Printing and Comparing
  - 9.10 Practice Modelling a Library System
- **10 Working with Libraries and Packages** — 10 classes
  - 10.1 Installing Packages with a Package Manager
  - 10.2 Managing Project Requirements
  - 10.3 Dates and Times
  - 10.4 Random Numbers and Simulations
  - 10.5 Working with Spreadsheets Programmatically
  - 10.6 Making Web Requests
  - 10.7 Reading Data from Web Services
  - 10.8 Command Line Arguments
  - 10.9 Reading Library Documentation
  - 10.10 Practice Building a Weather Lookup Tool
- **11 Automation Scripts for Everyday Office Tasks** — 10 classes
  - 11.1 Finding Tasks Worth Automating
  - 11.2 Renaming and Organising Files
  - 11.3 Batch Processing Documents
  - 11.4 Generating Reports Automatically
  - 11.5 Sending Automated Emails
  - 11.6 Scheduling Scripts to Run
  - 11.7 Collecting Data from Web Pages Responsibly
  - 11.8 Logging Script Activity
  - 11.9 Making Scripts Safe to Re-Run
  - 11.10 Practice Automating a Weekly Report
- **12 Python Practical Project** — 10 classes
  - 12.1 Choosing a Project Idea
  - 12.2 Writing Requirements and a Plan
  - 12.3 Designing Functions and Classes
  - 12.4 Building the Core Features
  - 12.5 Adding File or Data Storage
  - 12.6 Adding Error Handling
  - 12.7 Testing the Application
  - 12.8 Writing a Readme and User Guide
  - 12.9 Sharing Code with Version Control
  - 12.10 Presenting and Reviewing the Project

2 **Software Design, Architecture and Testing** 18 chapters · 120 classes **100 marks**

- **1 Understanding Core Software Design Principles** — 5 classes
  - 1.1 Introduction to Core Software Design Principles
  - 1.2 Exploring Abstraction in Software Design
  - 1.3 Understanding Encapsulation and Its Benefits
  - 1.4 Implementing Modular Design for Flexibility
  - 1.5 Applying Principles to Real-World Scenarios
- **2 Design Patterns and Their Applications** — 5 classes
  - 2.1 Understanding Design Patterns: An Overview
  - 2.2 Exploring Creational Patterns: Building Through Instantiation
  - 2.3 Harnessing Structural Patterns: Organizing Objects and Classes
  - 2.4 Implementing Behavioral Patterns: Managing Communication and Workflow
  - 2.5 Applying Design Patterns: Real-World Examples and Best Practices
- **3 Principles of Object-Oriented Design** — 5 classes
  - 3.1 Understanding the Core Concepts of Object-Oriented Design
  - 3.2 Identifying and Defining Classes and Objects
  - 3.3 Exploring Encapsulation and Information Hiding
  - 3.4 Mastering Inheritance for Code Reusability
  - 3.5 Implementing Polymorphism for Flexible Software Design
- **4 Designing for Scalability and Maintainability** — 5 classes
  - 4.1 Understanding Scalability and Maintainability Concepts
  - 4.2 Analyzing System Requirements for Scalability
  - 4.3 Implementing Design Patterns for Maintainability
  - 4.4 Evaluating Design Decisions for Scalability
  - 4.5 Applying Modular Design for Future Growth
- **1 Understanding Core Security Concepts in Software Development** — 6 classes
  - 1.1 Exploring the Fundamentals of Security in Software Development
  - 1.2 Understanding Threat Landscapes: Identifying Risks in Development
  - 1.3 Implementing Authentication and Authorization Mechanisms
  - 1.4 Securing Data: Techniques for Data Protection and Encryption
  - 1.5 Applying Secure Coding Practices to Prevent Vulnerabilities
  - 1.6 Conducting Security Testing and Audits in Software Projects
- **2 Implementing Secure Coding Practices** — 6 classes
  - 2.1 Understanding Secure Coding Principles
  - 2.2 Identifying Common Vulnerabilities in Code
  - 2.3 Implementing Input Validation Techniques
  - 2.4 Applying Authentication and Authorization Protocols
  - 2.5 Utilizing Secure Data Encryption Methods
  - 2.6 Conducting Code Reviews for Security Flaws
- **3 Advanced Threat Modeling and Risk Assessment** — 6 classes
  - 3.1 Understanding Advanced Threat Modeling Concepts
  - 3.2 Identifying Potential Threats in Software Systems
  - 3.3 Analyzing Threats with Risk Assessment Techniques
  - 3.4 Designing Effective Mitigation Strategies for Threats
  - 3.5 Applying Threat Modeling in Full-Stack Development
  - 3.6 Conducting Continuous Risk Assessments in API Integration
- **4 Integrating Security into DevOps Processes** — 6 classes
  - 4.1 Understanding the Importance of Security in DevOps
  - 4.2 Identifying Common Security Risks in DevOps
  - 4.3 Integrating Security Practices in Continuous Integration
  - 4.4 Implementing Automated Security Testing in DevOps
  - 4.5 Monitoring Security in Continuous Deployment
  - 4.6 Building a Security-Focused DevOps Culture
- **5 Testing and Monitoring for Security Vulnerabilities** — 6 classes
  - 5.1 Understanding Security Vulnerabilities in Software
  - 5.2 Identifying Common Security Threats in Code
  - 5.3 Implementing Unit Testing for Security
  - 5.4 Conducting Penetration Testing for Web Applications
  - 5.5 Utilizing Automated Tools for Vulnerability Scanning
  - 5.6 Monitoring Application Logs for Security Indicators
- **1 Understanding Product Lifecycle Stages** — 6 classes
  - 1.1 Explore the Concept of Product Lifecycle Stages
  - 1.2 Identify Key Characteristics of Each Lifecycle Stage
  - 1.3 Analyze the Introduction Stage Strategies
  - 1.4 Investigate Growth Stage Challenges and Opportunities
  - 1.5 Evaluate Strategies for the Maturity Stage
  - 1.6 Plan for the Decline Stage and Extension Tactics
- **2 Strategizing Product Development and Market Entry** — 6 classes
  - 2.1 Understanding the Product Development Cycle
  - 2.2 Identifying Market Opportunities for New Products
  - 2.3 Crafting a Strategic Product Development Plan
  - 2.4 Analyzing Competitive Landscape for Market Entry
  - 2.5 Developing a Go-To-Market Strategy
  - 2.6 Evaluating Product Development Success Metrics
- **3 Implementing Agile Practices in Product Lifecycle Management** — 6 classes
  - 3.1 Understanding Agile Principles in Product Lifecycle Management
  - 3.2 Identifying Key Agile Roles and Responsibilities
  - 3.3 Mapping Agile Methodologies to Product Stages
  - 3.4 Implementing Agile Tools for Product Development
  - 3.5 Facilitating Agile Collaboration in Cross-Functional Teams
  - 3.6 Evaluating Agile Success Metrics in Product Lifecycles
- **4 Optimizing Product Performance and User Experience** — 6 classes
  - 4.1 Understanding User Experience Fundamentals
  - 4.2 Analyzing Product Performance Metrics
  - 4.3 Identifying User Pain Points through Feedback
  - 4.4 Implementing Performance Optimization Techniques
  - 4.5 Enhancing User Experience with Design Improvements
  - 4.6 Evaluating Impact of Optimizations on User Satisfaction
- **5 Ensuring Sustainable Product Retirement and Transition** — 6 classes
  - 5.1 Understanding Sustainable Product Retirement
  - 5.2 Analyzing Environmental Impacts of Product Disposal
  - 5.3 Identifying Opportunities for Product Reuse and Recycling
  - 5.4 Developing a Transition Plan for Product Lines
  - 5.5 Implementing End-of-Life Strategies for Legacy Products
  - 5.6 Measuring Success in Product Retirement and Transition
- **15 Software Requirements and User Stories** — 10 classes
  - 15.1 Why Requirements Matter
  - 15.2 Functional and Non-Functional Requirements
  - 15.3 Identifying Stakeholders
  - 15.4 Interviews and Workshops for Requirements
  - 15.5 Writing User Stories
  - 15.6 Acceptance Criteria
  - 15.7 Prioritising Requirements
  - 15.8 Requirements Traceability
  - 15.9 Reviewing and Baselining Requirements
  - 15.10 Managing Requirement Changes
- **16 System Modelling with UML Diagrams** — 10 classes
  - 16.1 Purpose of Visual Modelling
  - 16.2 Use Case Diagrams
  - 16.3 Activity Diagrams for Workflows
  - 16.4 Class Diagrams and Relationships
  - 16.5 Sequence Diagrams for Interactions
  - 16.6 State Diagrams for Behaviour
  - 16.7 Entity Relationship Diagrams
  - 16.8 Wireframes and Screen Flows
  - 16.9 Choosing the Right Diagram
  - 16.10 Modelling a Small Booking System
- **17 Software Testing Techniques and Test Cases** — 10 classes
  - 17.1 Principles of Software Testing
  - 17.2 Levels of Testing
  - 17.3 Black Box Test Design
  - 17.4 Equivalence Partitioning and Boundary Values
  - 17.5 White Box Testing Basics
  - 17.6 Writing Unit Tests

3 **Machine Learning Fundamentals** 20 chapters · 120 classes **100 marks**

- **1 Understanding the Basics of Machine Learning** — 6 classes
  - 1.1 Defining Machine Learning: Concepts and Scope
  - 1.2 Exploring Types of Machine Learning: Supervised, Unsupervised, and Reinforcement
  - 1.3 Understanding Data: Inputs, Outputs, and Importance
  - 1.4 Recognizing Patterns: How Algorithms Learn
  - 1.5 Evaluating Models: Accuracy, Precision, and Recall
  - 1.6 Applying Ethical Considerations in Machine Learning
- **2 Data Preprocessing Techniques for Machine Learning** — 6 classes
  - 2.1 Understanding the Importance of Data Preprocessing
  - 2.2 Exploring Data Cleaning Methods
  - 2.3 Handling Missing Data Effectively
  - 2.4 Normalizing and Scaling Data for Consistency
  - 2.5 Encoding Categorical Variables for Model Readiness
  - 2.6 Implementing Feature Engineering Techniques
- **3 Exploring Supervised Learning Algorithms** — 6 classes
  - 3.1 Understanding Supervised Learning Concepts
  - 3.2 Exploring Linear Regression Techniques
  - 3.3 Implementing Decision Trees for Classification
  - 3.4 Analyzing Support Vector Machines
  - 3.5 Evaluating Model Performance Metrics
  - 3.6 Applying Ensemble Methods in Supervised Learning
- **4 Unsupervised Learning and Clustering Techniques** — 6 classes
  - 4.1 Understanding Unsupervised Learning
  - 4.2 Exploring Clustering Techniques
  - 4.3 Implementing K-Means Clustering
  - 4.4 Analyzing Hierarchical Clustering
  - 4.5 Applying DBSCAN for Density-Based Clustering
  - 4.6 Evaluating Clustering Performance
- **5 Model Evaluation and Performance Metrics** — 6 classes
  - 5.1 Understanding the Importance of Model Evaluation
  - 5.2 Exploring Common Performance Metrics in Machine Learning
  - 5.3 Comparing Different Model Evaluation Techniques
  - 5.4 Interpreting Confusion Matrix for Classification Models
  - 5.5 Calculating Precision, Recall, and F1-Score
  - 5.6 Assessing Model Performance Through ROC Curves and AUC
- **1 Understanding Model Evaluation Metrics** — 6 classes
  - 1.1 Exploring the Importance of Model Evaluation in AI
  - 1.2 Understanding Accuracy, Precision, and Recall Metrics
  - 1.3 Interpreting F1 Score and Its Application
  - 1.4 Analyzing ROC Curves and AUC in Model Assessment
  - 1.5 Evaluating Confusion Matrix Insights for Model Improvement
  - 1.6 Implementing Cross-Validation for Robust Model Evaluation
- **2 Advanced Techniques for Model Validation** — 6 classes
  - 2.1 Understanding Cross-Validation Techniques
  - 2.2 Implementing k-Fold Cross-Validation
  - 2.3 Applying Leave-One-Out Cross-Validation
  - 2.4 Utilizing Stratified Sampling in Validation
  - 2.5 Analyzing Model Validation Metrics
  - 2.6 Comparing Validation Approaches for Robust Models
- **3 Hyperparameter Tuning and Optimization** — 6 classes
  - 3.1 Understanding Hyperparameters in Machine Learning Models
  - 3.2 Analyzing the Impact of Hyperparameter Values
  - 3.3 Exploring Methods for Hyperparameter Tuning
  - 3.4 Implementing Grid Search for Hyperparameter Optimization
  - 3.5 Applying Random Search for Enhanced Model Performance
  - 3.6 Comparing Hyperparameter Optimization Techniques
- **4 Model Deployment Strategies** — 6 classes
  - 4.1 Understanding Model Deployment Environments
  - 4.2 Preparing Models for Deployment
  - 4.3 Containerizing Machine Learning Models
  - 4.4 Implementing Deployment Pipelines
  - 4.5 Monitoring and Managing Deployed Models
  - 4.6 Scaling Model Deployments Across Platforms
- **5 Monitoring and Maintaining Deployed Models** — 6 classes
- **1 Deep Learning Architectures** — 6 classes
  - 1.1 Explaining Neural Network Fundamentals
  - 1.2 Constructing Convolutional Neural Networks
  - 1.3 Designing Recurrent Neural Networks
  - 1.4 Implementing LSTM and GRU Networks
  - 1.5 Exploring Transformer Architectures
  - 1.6 Applying Deep Learning Architectures to Real-world Problems
- **2 Natural Language Processing Techniques** — 6 classes
  - 2.1 Understanding the Basics of Natural Language Processing
  - 2.2 Exploring Tokenization and Text Preprocessing
  - 2.3 Analyzing Syntax with Part-of-Speech Tagging
  - 2.4 Utilizing Named Entity Recognition for Data Extraction
  - 2.5 Applying Sentiment Analysis Techniques
  - 2.6 Implementing Language Models for Text Generation
- **3 Generative Adversarial Networks (GANs)** — 6 classes
  - 3.1 Understanding the Basics of GANs
  - 3.2 Exploring the Architecture of Generator and Discriminator
  - 3.3 Training GANs: Techniques and Challenges
  - 3.4 Implementing GANs Using Popular Frameworks
  - 3.5 Evaluating GAN Performance Metrics
  - 3.6 Applying GANs to Real-World Scenarios
- **4 AI in Reinforcement Learning Systems** — 6 classes
  - 4.1 Understanding Reinforcement Learning: Fundamentals and Concepts
  - 4.2 Exploring Key Algorithms in Reinforcement Learning
  - 4.3 Analyzing Policy and Value Functions in RL
  - 4.4 Implementing Deep Reinforcement Learning Techniques
  - 4.5 Evaluating and Optimizing Reward Structures
  - 4.6 Applying Reinforcement Learning to Real-World Scenarios
- **5 Ethical and Responsible AI** — 6 classes
  - 5.1 Understanding the Principles of Ethical AI
  - 5.2 Exploring Bias and Fairness in AI Systems
  - 5.3 Analyzing AI Transparency and Explainability
  - 5.4 Ensuring Privacy and Data Protection in AI
  - 5.5 Evaluating the Societal Impact of AI Technologies
  - 5.6 Implementing Ethical Guidelines in AI Development
- **1 Introduction to Machine Learning** — 6 classes
  - 1.1 Fundamentals of Machine Learning
  - 1.2 Types of Machine Learning
  - 1.3 Supervised vs Unsupervised Learning
  - 1.4 Applications in Hospitality
  - 1.5 Data Preparation Techniques
  - 1.6 Ethics in Machine Learning
- **2 Predictive Analytics Techniques** — 6 classes
  - 2.1 Understanding Predictive Analytics
  - 2.2 Regression Analysis Basics
  - 2.3 Time Series Forecasting
  - 2.4 Classification Methods Overview
  - 2.5 Model Evaluation Metrics
  - 2.6 Real-World Applications
- **3 Customer Preference Modeling** — 6 classes
  - 3.1 Understanding Customer Preferences
  - 3.2 Data Collection Techniques
  - 3.3 Segmentation Strategies
  - 3.4 Behavioral Analysis Methods
  - 3.5 Personalization Techniques
  - 3.6 Feedback Loop Integration
- **4 Marketing Strategy Development** — 6 classes
  - 4.1 Defining Marketing Objectives
  - 4.2 Target Audience Identification
  - 4.3 Digital Marketing Strategies
  - 4.4 Content Marketing Approaches
  - 4.5 Campaign Planning Essentials
  - 4.6 Budgeting for Marketing
- **5 Evaluating Marketing Campaigns** — 6 classes

4

AI Solution Design, Deployment and Ethics

18 chapters · 120 classes **100 marks**

- **1 Understanding the Principles of AI Solution Design** — 6 classes
  - 1.1 Exploring the Fundamentals of AI Solution Design
  - 1.2 Identifying Core Components of AI Systems
  - 1.3 Analyzing Data Requirements for AI Solutions
  - 1.4 Prioritizing Objectives for Effective AI Design
  - 1.5 Evaluating Ethical Considerations in AI Solutions
  - 1.6 Applying AI Design Principles to Real-World Scenarios
- **2 Data Acquisition and Preprocessing Strategies** — 6 classes
  - 2.1 Understanding Data Acquisition Methods
  - 2.2 Identifying Relevant Data Sources
  - 2.3 Implementing Data Collection Techniques
  - 2.4 Cleaning and Preparing Raw Data
  - 2.5 Applying Data Transformation Techniques
  - 2.6 Evaluating Data Quality for AI Solutions
- **3 Model Selection and Evaluation Techniques** — 6 classes
  - 3.1 Distinguishing Models: An Introduction to Supervised and Unsupervised Learning
  - 3.2 Exploring Model Selection: Criteria and Considerations
  - 3.3 Applying Cross-Validation Techniques for Model Assessment
  - 3.4 Evaluating Model Performance: Metrics and Interpretations
  - 3.5 Selecting Optimal Models: Understanding Bias-Variance Trade-off
  - 3.6 Implementing Model Evaluation: Techniques and Tools
- **4 Deployment and Integration of AI Models** — 6 classes
  - 4.1 Exploring AI Model Deployment Environments
  - 4.2 Preparing AI Models for Deployment
  - 4.3 Implementing Continuous Integration and Continuous Deployment (CI/CD) in AI
  - 4.4 Integrating AI Models with Existing Systems
  - 4.5 Monitoring and Maintaining Deployed AI Models
  - 4.6 Evaluating the Performance of Integrated AI Solutions
- **5 Monitoring and Optimizing AI Solutions** — 6 classes
  - 5.1 Understanding Key Metrics for AI Monitoring
  - 5.2 Implementing Continuous Performance Tracking
  - 5.3 Analyzing AI Solution Data for Insights
  - 5.4 Utilizing Feedback Loops to Optimize AI Models
  - 5.5 Identifying and Mitigating AI Drift
  - 5.6 Deploying Tools for Automated AI System Alerts
- **1 Overview of AI in Hospitality** — 6 classes
  - 1.1 Introduction to AI Technologies
  - 1.2 Historical Context of AI in Hospitality
  - 1.3 Current Trends in AI Applications
  - 1.4 Key AI Tools for Hospitality
  - 1.5 AI Ethics in Hospitality
  - 1.6 Future of AI in Hospitality
- **2 Optimizing Service Delivery** — 6 classes
  - 2.1 AI-Driven Customer Insights
  - 2.2 Personalization Techniques Using AI
  - 2.3 Automation in Service Processes
  - 2.4 Enhancing Guest Experience with AI
  - 2.5 Real-Time Service Optimization
  - 2.6 Feedback Mechanisms and AI
- **3 AI Workflow Management** — 6 classes
  - 3.1 Designing AI Workflows
  - 3.2 Integrating AI with Existing Systems
  - 3.3 Monitoring AI Performance
  - 3.4 Collaboration Tools for AI Teams
  - 3.5 Managing AI Project Lifecycles
  - 3.6 Troubleshooting AI Workflows
- **4 Risk Management in AI Solutions** — 6 classes
  - 4.1 Identifying AI Risks
  - 4.2 Mitigation Strategies for AI Risks
  - 4.3 Compliance and Regulatory Considerations
  - 4.4 Data Privacy and Security in AI
  - 4.5 Crisis Management in AI Implementation
  - 4.6 Evaluating Risk Management Frameworks
- **5 Evaluating AI Impact on Operations** — 6 classes
  - 5.1 Measuring AI Success Metrics
  - 5.2 Impact Assessment Techniques
  - 5.3 ROI Analysis for AI Investments
  - 5.4 Feedback Loop for Continuous Improvement
  - 5.5 Case Studies on AI Impact
  - 5.6 Strategic Planning Based on AI Insights
- **1 Vision and Strategic Planning** — 6 classes
  - 1.1 Crafting a Compelling Vision
  - 1.2 Setting Strategic Objectives
  - 1.3 Analyzing Market Trends
  - 1.4 Developing Actionable Strategies
  - 1.5 Measuring Success Metrics
  - 1.6 Adapting to Change
- **2 Inspiring Teams for Success** — 6 classes
  - 2.1 Building Trust and Rapport
  - 2.2 Empowering Team Members
  - 2.3 Effective Communication Techniques
  - 2.4 Recognizing and Rewarding Achievements
  - 2.5 Fostering a Collaborative Environment
  - 2.6 Navigating Conflict Resolution
- **3 Aligning Values and Mission** — 6 classes
  - 3.1 Defining Core Values
  - 3.2 Integrating Values into Culture
  - 3.3 Communicating the Mission Effectively
  - 3.4 Creating a Value-Driven Organization
  - 3.5 Evaluating Mission Alignment
  - 3.6 Incorporating Stakeholder Perspectives
- **4 Driving Growth and Sustainability** — 6 classes
  - 4.1 Identifying Growth Opportunities
  - 4.2 Sustainable Business Practices
  - 4.3 Leveraging Technology for Growth
  - 4.4 Measuring Long-Term Impact
  - 4.5 Strategic Partnerships for Growth
  - 4.6 Balancing Profit and Purpose
- **5 Cultivating Innovation and Improvement** — 6 classes
  - 5.1 Encouraging Creative Thinking
  - 5.2 Implementing Continuous Improvement
  - 5.3 Utilizing Feedback for Growth
  - 5.4 Exploring Emerging Technologies
  - 5.5 Creating an Innovation Culture
  - 5.6 Measuring Innovation Outcomes
- **16 Responsible and Ethical AI Governance in Practice** — 10 classes
  - 16.1 Principles of Responsible AI
  - 16.2 Bias and Fairness in Everyday AI Tools
  - 16.3 Privacy and Consent When Using AI
  - 16.4 Transparency and Explainability
  - 16.5 Accountability and Human Oversight
  - 16.6 Writing a Simple AI Use Policy
  - 16.7 AI Impact Assessment Steps
  - 16.8 Logging and Auditing AI Decisions
  - 16.9 Handling AI Incidents and Complaints
  - 16.10 Reviewing an AI Policy for Compliance
- **17 Generative AI Tools for Workplace Productivity** — 10 classes
  - 17.1 What Generative AI Can and Cannot Do
  - 17.2 Choosing the Right Tool for a Task
  - 17.3 Writing Clear Prompts
  - 17.4 Drafting Documents and Emails with AI
  - 17.5 Summarising Reports and Meetings
  - 17.6 Generating Ideas and Outlines
  - 17.7 Checking AI Output for Accuracy
  - 17.8 Avoiding Confidential Data in Prompts
  - 17.9 Citing and Disclosing AI Assistance
  - 17.10 Building a Personal AI Workflow
- **18 Planning and Delivering an AI Pilot Project** — 10 classes

LAPT — 85 Great Portland Street, W1W 7LT, London, United Kingdom
info@lapt.org • +44 7513 283044 • lapt.org

UKPRN 10067351 • Company 4984798
Curriculum current at the date of issue.