

月日	日数	研修時間	研修時間	昼休憩	カリキュラム	詳細
4/9	1	3	09:00～12:00	12:00～13:00	オリエンテーション & マインドセット構築	【グループワーク】 チームビルディングとエンジニアマインドの確立 実施内容: チームを編成し、「AIに使われるコーダー」と「AIを指揮するソフトウェアエンジニア」の違いを議論し、人間主導のスタンスを共有します。 習得スキル: チームビルディング能力、人間主導（当事者意識）マインド 講義： コンピュータサイエンスの全体像（基礎理論とハードウェア） 実施内容:データが処理される根本的な流れ（入力→処理→出力）を押さえた上で、2進数・16進数やデータ量の単位などの基礎理論、五大装置やキャッシュメモリ等のハードウェアの仕組み、およびOS・ミドルウェア・アプリケーションのレイヤー構造を俯瞰します。 習得スキル: ITシステム全体構造の把握力、ITパスポートレベルの基礎理論・ハードウェア知識
		4.5	13:00-17:30	—		
4/10	休講日					
4/11						
4/12	1	3	09:00～12:00	12:00～13:00	AI駆動開発における 論理的思考力	講義・演習：AI駆動開発における論理的思考力 実施内容: AIに適切な指示（プロンプト）を出す前提となる「システム開発思考」を学びます。漠然としたビジネス課題やシステム要件を「背景」「目的」「制約条件」「入力と出力」などの構成要素に分解し、MECE（モレなくダブリなく）の視点を用いて要件を構造化する論理的思考トレーニングを行います。構造化された要件からAIへのコード生成指示へ繋ぐため、人間がシステム全体の「処理のステップ」を論理的に組み立てる演習を行います。曖昧な自然言語を排除し、処理の分岐や反復をフローチャートや論理的な疑似コード（疑似言語）に変換することで、AIに誤解なく伝わるプロンプトを理解します。論理的に組み立てた要件と処理フローを、AIが正確に解釈・実行できる「構造化プロンプト」へと落とし込みます。 さらに、AIの出力結果をクリティカルに評価し、意図した挙動にならない場合の「原因究明→仮説立案→プロンプトの論理的修正」という仮説検証サイクルを実践します。 習得スキル: システム開発思考、複雑な要件の構造化能力、MECEによる要素分解、アルゴリズム的思考、処理フローの論理構築力、疑似コード（構造化された自然言語）の作成能力、構造化プロンプトの設計力、出力結果に対するクリティカルシンキング、仮説検証によるトラブルシューティング能力
		4.5	13:00-17:30	—		
4/13	1	3	09:00～12:00	12:00～13:00	AI駆動開発における 論理的思考力	同上
		4.5	13:00-17:30	—		
4/14	1	3	09:00～12:00	12:00～13:00	ITアーキテクチャ概論とWebの仕組み	講義：Webアプリケーションの基本アーキテクチャ 実施内容: クライアント/サーバーモデル、URLの構造、HTTP/HTTPS（暗号化通信）の違い、HTTPメソッド（GET / POSTの違い）を学びます。 習得スキル: Webシステムの基本構造と通信プロトコルの理解 使用プログラミング言語: なし 使用ツール: なし 【個人ワーク】 ブラウザの開発者ツールを用いた通信の観察 実施内容: 実際にWebサイトにアクセスし、HTTPリクエスト/レスポンスの中身（ヘッダー、ボディ、ステータスコード）を確認します。 習得スキル: HTTPプロトコルの実践的理解、開発者ツールの基本操作 使用プログラミング言語: なし 使用ツール: Webブラウザ（開発者ツール）
		4.5	13:00-17:30	—		

4/15	1	3	09:00～12:00	12:00～13:00	ネットワーク基礎とインフラストラクチャ	講義：TCP/IP、DNS、インターネットの仕組み 実施内容： パケットの送受信から名前解決まで、インターネットを支えるネットワークの基盤技術を構造的に理解します。 習得スキル： ネットワークプロトコルの基礎知識 【グループワーク】ネットワーク構成図の作成 実施内容： 与えられた要件に基づき、Webシステムのネットワーク構成図を描画します。 習得スキル： インフラ構成の図解・合意形成力
		4.5	13:00～17:30	—		
4/16	1	3	09:00～12:00	12:00～13:00	ネットワーク基礎とインフラストラクチャ	同上
		4.5	13:00～17:30	—		
4/17	休講日					
4/18						
4/19	1	3	09:00～12:00	12:00～13:00	データベース基礎とデータモデリング	講義：RDBMSの基本とデータ形式 実施内容： テーブルやレコード、トランザクション（ACID特性）を学びます。またJSON、CSV、Markdown等、システム間でやり取りされるデータ形式の特徴を理解します。 習得スキル： RDBMSの基本概念と各種データフォーマットの理解 【グループワーク】正規化とER図の作成・SQL演習 実施内容： データの冗長性を排除する正規化とER図の作成を行います。またテスト用DBに対しSQLクエリを記述しデータを抽出します。 習得スキル： データベース論理設計、SQLの記述力 使用プログラミング言語： SQL 使用ツール： DBクライアントツール
		4.5	13:00～17:30	—		
4/20	1	3	09:00～12:00	12:00～13:00	データベース基礎とデータモデリング	同上
		4.5	13:00～17:30	—		
4/21	1	3	09:00～12:00	12:00～13:00	データベース基礎とデータモデリング	同上
		4.5	13:00～17:30	—		
4/22	1	3	09:00～12:00	12:00～13:00	【AI制限区間】アルゴリズムとデータ構造	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：基礎～高度なデータ構造とアルゴリズム 実施内容： 配列・リストから木構造・グラフ等まで俯瞰。ソート・探索の自力実装と、非効率なコードをリファクタリングして計算量を減らす演習を行います。 条件分岐の基礎となる論理演算（AND、OR、NOT、XOR）とベン図の概念を学び、複雑な条件を論理的に組み立てる思考法を習得します。 習得スキル： 計算量（オーダー）の概念、アルゴリズム実装とコード最適化能力 論理演算の理解、条件式の組み立て能力 使用プログラミング言語： Scratch 使用ツール： Scratch
		4.5	13:00～17:30	—		
4/23	1	3	09:00～12:00	12:00～13:00	【AI制限区間】アルゴリズムとデータ構造	同上
		4.5	13:00～17:30	—		
4/24	休講日					

4/25						
4/26	1	3	09:00～12:00	12:00～13:00	【AI制限区間】アルゴリズムとデータ構造	同上
		4.5	13:00-17:30	—		
4/27	1	3	09:00～12:00	12:00～13:00	【AI制限区間】Python基礎	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：Pythonプログラミング基礎 実施内容: 変数の宣言、データ型、条件分岐（if文）、繰り返し処理（for/while）、基本的な関数の定義を学びます。 習得スキル: プログラミング言語の基礎構文の理解、順次・分岐・反復のロジック構築力 講義・演習：コレクション 実施内容: リスト、辞書（ディクショナリ）、タプル等のデータ構造の操作を学びます。 習得スキル: コレクションの操作 使用プログラミング言語: Python 使用ツール: VS Code
		4.5	13:00-17:30	—		
4/28	1	3	09:00～12:00	12:00～13:00	【AI制限区間】Python基礎	同上
		4.5	13:00-17:30	—		
4/29	休講日					
4/30	1	3	09:00～12:00	12:00～13:00	【AI制限区間】Python応用	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：オブジェクト指向の概念と実装 実施内容: クラスとインスタンス、メソッド、継承など、オブジェクト指向の概念を用いて実装し、コードの再利用性を学びます。 習得スキル: オブジェクト指向の概念理解とクラス設計力 使用プログラミング言語: Python 使用ツール: VS Code 講義・演習：ライブラリ活用と入出力 実施内容: 外部ライブラリ（モジュール）のインポート、仮想環境（venv）の利用、およびCSVやJSONファイルの読み書き処理を実装します。 習得スキル: ライブラリ管理（pip等）、ファイル入出力操作 使用プログラミング言語: Python 使用ツール: VS Code 講義・演習：エラーハンドリング 実施内容: エラーを捕捉する例外処理（try-except）を学びます。 習得スキル: 堅牢なエラーハンドリング 使用プログラミング言語: Python 使用ツール: VS Code
		4.5	13:00-17:30	—		
合計	15	108				

月日	日数	研修時間	研修時間	昼休憩	カリキュラム	詳細
5/1	休講日					
5/2						
5/3						
5/4						
5/5						
5/6	1	3	09:00～12:00	12:00～13:00	【AI制限区間】 Python応用	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：オブジェクト指向の概念と実装 実施内容: クラスとインスタンス、メソッド、継承など、オブジェクト指向の概念を用いて実装し、コードの再利用性を学びます。 習得スキル: オブジェクト指向の概念理解とクラス設計力 使用プログラミング言語: Python 使用ツール: VS Code 講義・演習：ライブラリ活用と入出力 実施内容: 外部ライブラリ（モジュール）のインポート、仮想環境（venv）の利用、およびCSVやJSONファイルの読み書き処理を実装します。 習得スキル: ライブラリ管理（pip等）、ファイル入出力操作 使用プログラミング言語: Python 使用ツール: VS Code 講義・演習：エラーハンドリング 実施内容: エラーを捕捉する例外処理（try-except）を学びます。 習得スキル: 堅牢なエラーハンドリング 使用プログラミング言語: Python 使用ツール: VS Code
		4.5	13:00-17:30	—		
5/7	1	3	09:00～12:00	12:00～13:00	【AI制限区間】 Python応用	同上
		4.5	13:00-17:30	—		
5/8	休講日					
5/9						
5/10	1	3	09:00～12:00	12:00～13:00	【AI制限区間】 HTML/CSS基礎	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：HTMLの文書構造とCSSスタイリング 実施内容: Webページの骨格となるHTMLのタグ構造と、CSSを用いた基本的なスタイリング、Flexbox/Gridによるモダンなレイアウト作成を学びます。 習得スキル: HTML/CSSの基本構文、Webページの静的コーディング・レイアウト構築力 使用プログラミング言語: HTML, CSS 使用ツール: VS Code, Webブラウザ
		4.5	13:00-17:30	—		

5/11	1	3	09:00～12:00	12:00～13:00	【AI制限区間】 JavaScript基礎	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：JavaScriptプログラミング基礎 実施内容： 変数（let/const）、関数（アロー関数）、配列とオブジェクトの操作など、JavaScript独自の文法をPythonと比較しながら学びます。 習得スキル： モダンなJavaScript文法の理解 使用プログラミング言語： JavaScript 使用ツール： VS Code, Webブラウザ
		4.5	13:00～17:30	—		
5/12	1	3	09:00～12:00	12:00～13:00	【AI制限区間】 JavaScript基礎	※ルール：AIコーディングツールの利用を一切禁止 講義・演習：フロントエンドの動的UI構築 実施内容： HTML/CSSと連携し、JavaScriptからブラウザの要素（DOM）を動的に書き換えたり、クリック等のイベントを取得する処理を実装します。 習得スキル： フロントエンドの動的UI構築スキル 使用プログラミング言語： JavaScript, HTML, CSS 使用ツール： VS Code, Webブラウザ
		4.5	13:00～17:30	—		
5/13	1	3	09:00～12:00	12:00～13:00	【AI制限区間】 JavaScript応用（API通信）	※ルール：AIコーディングツールの利用を一切禁止 講義・演習： Fetch API 実施内容： Web開発で頻きやすいAPIを理解し、Fetch APIを用いて外部からJSONデータを取得します。 習得スキル： Web API通信の実装力 使用プログラミング言語： JavaScript 使用ツール： VS Code, Webブラウザ
		4.5	13:00～17:30	—		
5/14	1	3	09:00～12:00	12:00～13:00	FlaskによるWebアプリケーション開発	※ルール：AIコーディングツールの利用を一切禁止 講義・演習： FlaskによるWebアプリケーション開発 Webの仕組みとFlaskの基本操作 セッション管理とデータベース連携 実施内容： Webの基本構造やHTTP通信について学習し、Flaskを使用してルーティング、リクエスト・レスポンス処理、テンプレートによる面 面表示を実装します。 Cookieとセッションによる状態管理、MVCの考え方、Flask-SQLAlchemyを使用したデータの登録・検索・更新・削除を実装しま す。 習得スキル： Webアプリケーションの基礎知識、Flaskを用いた基本的な画面・リクエスト処理 セッション管理、MVCを意識したアプリケーション設計、ORMによるデータベース操作 使用プログラミング言語： Python, HTML、Python, HTML, SQL 使用ツール： VS Code, Flask, Jinja2、VS Code, Flask, Flask-SQLAlchemy, Flask-Migrate
		4.5	13:00～17:30	—		

5/15	休講日					
5/16						
5/17	1	3	09:00～12:00	12:00～13:00	FlaskによるWebアプリケーション開発	同上
		4.5	13:00～17:30	—		
5/18	1	3	09:00～12:00	12:00～13:00	第1フェーズ評価ゲート（実践デバッグ & リバースエンジニアリング）	【個人ワーク】 バグハント・タイムアタック（実践力評価） 実施内容: 意図的に「論理的欠陥（無限ループなど）」「非効率なデータ構造」が埋め込まれたソースコードを渡し、制限時間内にAIを使わず自力でバグを特定・修正させます。修正箇所とその理由はコメントで記述させます。 習得スキル: コードリーディング、実践的なデバッグ能力、脆弱性の発見と修正 使用プログラミング言語: Python または JavaScript 使用ツール: VS Code
		4.5	13:00～17:30	—		
5/19	1	3	09:00～12:00	12:00～13:00	セキュアコーディングと認証・暗号化	講義・演習：Flask脆弱性対策とセキュアコーディング 実施内容: Webアプリケーションで頻発する脆弱性の原理と対策を学びます。Flask環境における代表的な攻撃（SQLインジェクション、XSS、CSRF、セッションハイジャックなど）を理解した上で、Flask-WTF（CSRF保護）、Jinja2の自動エスケープ機構、SQLAlchemyのパラメタ化クエリ（プリペアドステートメント）を用いた防御手法を習得します。 習得スキル: Flask / SQLAlchemyにおける脆弱性の特定と防御策の実装力・CSRF対策（Flask-WTF）、XSS対策（Jinja2エスケープ）、SQLインジェクション対策・コード監査および脆弱性診断・修正能力 使用プログラミング言語: Python 使用ツール: VS Code 【個人ワーク】 リバースエンジニアリング & ドキュメンテーション（構造的理解の評価） 実施内容: ソースコードを見せずに「動いているシンプルなWebアプリ（ブラックボックス）」を触らせ、その挙動から裏側の「DB設計（ER図）」「ネットワーク構成図」「処理のフローチャート」を推測してドキュメントを作成・提出させます。 習得スキル: システム全体の俯瞰力、要件推測・リバースエンジニアリング能力、ドキュメンテーション 使用プログラミング言語: なし 使用ツール: Webブラウザ、設計ツール
		4.5	13:00～17:30	—		
5/20	1	3	09:00～12:00	12:00～13:00	バージョン管理とSDLC・アジャイル基礎	講義・演習：Git/GitHub基礎、SDLCとスクラム開発 実施内容: Gitの基本操作（コンフリクト解消含む）、SDLCの流れ、アジャイル開発のスクラムセレモニーとタスク分割までを学びます。 習得スキル: バージョン管理、SDLCの理解、アジャイル開発のタスク管理能力 使用プログラミング言語: なし 使用ツール: VS Code, Git
		4.5	13:00～17:30	—		

5/21	1	3	09:00～12:00	12:00～13:00	生成AIの裏側とプロンプトエンジニアリング	講義・演習：LLMの動作原理と構造化プロンプト 実施内容: LLMのハルシネーションの理由と「オプトアウト」の重要性を学び、AIに対する高精度な指示（構造化プロンプト）を設計・検証します。 習得スキル: LLMのリスク管理とプロンプト設計力 使用プログラミング言語: なし 使用ツール: GitHub Copilot
		4.5	13:00～17:30	—		
5/22	休講日					
5/23						
5/24	1	3	09:00～12:00	12:00～13:00	Difyを用いたAIワークフローとPoC構築体験	講義・演習：ノーコードによるAIアーキテクチャの俯瞰 実施内容: LLMの仕組みを学んだ直後に、Difyを使って「RAG（検索拡張生成）」や「条件分岐」を持つAIアプリをノーコードで構築し、AIアプリのアーキテクチャの全体像を体感します。 習得スキル: ノーコードAI開発ツール（Dify）の基本操作、AIアーキテクチャの俯瞰 使用プログラミング言語: なし 使用ツール: Dify
		4.5	13:00～17:30	—		
5/25	1	3	09:00～12:00	12:00～13:00	Difyを用いたAIワークフローとPoC構築体験	同上
		4.5	13:00～17:30	—		
5/26	1	3	09:00～12:00	12:00～13:00	コーディングAI操作基礎と高速開発	【個人ワーク】VS CodeとGitHub Copilotの活用とコード自動生成 実施内容: コードの自動生成、レガシーコードのリファクタリング、ドキュメント自動生成を実践します。 習得スキル: コーディングAIツールの実践活用、超高速開発スキル 使用プログラミング言語: Python または JavaScript 使用ツール: VS Code, GitHub Copilot
		4.5	13:00～17:30	—		
5/27	1	3	09:00～12:00	12:00～13:00	AIによるテスト設計とテストコード自動生成	講義・演習：テストケース抽出と pytest/Jest 生成 実施内容: AIに網羅的なテストケース（正常系・異常系・境界値）を洗い出させ、Python (pytest) または JavaScript (Jest) のテストコードを一気に生成してTDDの骨組みを構築します。 習得スキル: AIを用いたテスト設計・テストコード生成スキル、TDDの初動構築 使用プログラミング言語: Python または JavaScript 使用ツール: VS Code, GitHub Copilot, pytest / Jest
		4.5	13:00～17:30	—		

5/28	1	3	09:00～12:00	12:00～13:00	エラー駆動のAI実装/複雑なテストのAI活用	講義・演習：AIを用いた実装・デバッグサイクルとコード最適化 実施内容: テスト駆動開発の考え方に沿い、生成したテストが通るようなプログラムをAIに実装させます。テスト成功後、続けてコードの可読性やパフォーマンスを向上させるリファクタリングを行い、TDDの一連のサイクルを完遂させます。 習得スキル: AIを用いた実装・デバッグサイクル、エラー解読力、安全なリファクタリングスキル 使用プログラミング言語: Python または JavaScript 使用ツール: VS Code, GitHub Copilot, pytest / Jest 講義・演習：モック（Mock）の生成 実施内容: 外部API連携やDBアクセスのテストなど、つまずきやすい「モック」の作成をAIに任せ、独立した堅牢なテストを構築します。 習得スキル: モック（Mock）テストの理解 使用プログラミング言語: Python または JavaScript 使用ツール: VS Code, GitHub Copilot, pytest / Jest
		4.5	13:00~17:30	—		
5/29	休講日					
5/30						
5/31	1	3	09:00～12:00	12:00～13:00	API連携と外部システム統合（TDD実践）	講義・演習：REST API設計原則とシステム連携のTDD開発 実施内容: 外部API（天気や翻訳等）を呼び出して自システムに組み込む処理を、これまでに学んだTDDのサイクル（テスト作成→実装）に則って実装します。 習得スキル: API連携実装力、TDDを用いた外部サービス統合 使用プログラミング言語: Python または JavaScript 使用ツール: VS Code, GitHub Copilot, pytest / Jest
		4.5	13:00~17:30	—		
合計	18	135				

月日	日数	研修時間	研修時間	昼休憩	カリキュラム	詳細
6/1	1	3	09:00～12:00	12:00～13:00	フロントエンドとバックエンドの結合テスト	講義・演習：APIとUIの結合テストおよびAIを活用したデバッグ 実施内容： バックエンドAPIとフロントエンドUIを統合し、データの取得、登録、更新、エラー応答などを対象とした結合テストを実施します。テストの失敗やシステム統合時に発生したエラーについて、ログやエラーメッセージをAIに提示して原因を分析し、提案された修正方法の妥当性を確認しながら問題を解決する手法を学びます。 習得スキル： フロントエンドとバックエンドの結合テスト、エラー原因の切り分け、AIを活用したデバッグ 使用プログラミング言語： Python, JavaScript 使用ツール： VS Code, GitHub Copilot, pytest / Jest
		4.5	13:00～17:30	-		
6/2	1	3	09:00～12:00	12:00～13:00	CI/CD環境での自動テスト構築	講義・演習：CI/パイプライン構築、CDの基礎 実施内容： コードをプッシュすると自動でテスト（pytest/Jest）が実行される、継続的インテグレーション（CI）環境を構築します。 習得スキル： CI/パイプライン構築の基礎スキル 使用プログラミング言語： YAML, Python または JavaScript 使用ツール： VS Code, GitHub 講義・演習：開発フローへのAIレビュー統合 実施内容： CI/パイプラインにLLMを組み込み、「プルリクエスト作成時に、AIが自動でコードレビューのコメントを残す」仕組みを構築します。 習得スキル： 開発フローへのAI組み込み、モダンな開発環境の設計 使用プログラミング言語： YAML, Python または JavaScript 使用ツール： VS Code, Git, LLM API
		4.5	13:00～17:30	-		
6/3	1	3	09:00～12:00	12:00～13:00	第2フェーズ評価ゲート（AI駆動・TDD実装 & CI/CDパイプライン構築）	【個人ワーク】AI駆動・TDD実装タイムアタック（実践力評価） 実施内容： 新たな仕様変更の要件定義書を渡し、制限時間内にAIを活用してテスト駆動開発（TDD）のサイクルを回させます。AIにテストコード（pytest/Jest）を生成させ、「レッド」から「グリーン」へのエラー駆動実装、およびリファクタリングまでの全工程を自律的に完遂させます。 習得スキル： AIを用いた高速なエラー駆動実装力、テストコード生成スキル、安全なリファクタリング実践能力 使用プログラミング言語： Python または JavaScript 使用ツール： VS Code, GitHub Copilot, pytest / Jest 【個人ワーク】CI/CDパイプライン構築 & ドキュメンテーション（構造的理解の評価） 実施内容： 午前に実装したコードベースに対し、CI/CDパイプライン（自動テスト実行とAIコードレビュー自動化）を構築させます。さらに、構築したパイプラインの構成図と、AIレビューの動作ロジックを説明するドキュメントを作成・提出させます。（※基準未達者は補講） 習得スキル： CI/CDパイプラインの構築・運用能力、開発フローへのAI組み込みの構造的理解、ドキュメンテーション 使用プログラミング言語： YAML, Python または JavaScript 使用ツール： VS Code, Git, GitHub, LLM API, 図解ツール

		4.5	13:00～17:30	-		VS CODE, GIT, GITHUB, LLM API, 設計ツール
6/4	1	3	09:00～12:00	12:00～13:00	Linux入門	講義・演習：Linuxの基本知識、コマンド演習 実施内容： Webサイトを構成するサーバーのOSとして広く採用されているLinuxについて、操作の実習を行います。 習得スキル： Linuxの基本コマンド操作スキル、シェルの基礎知識、シェルスクリプトの作成スキル 使用プログラミング言語： なし 使用ツール： Terraterm、Virtualbox
		4.5	13:00～17:30	-		
6/5	休講日					
6/6	休講日					
6/7	1	3	09:00～12:00	12:00～13:00	Linux入門	同上
		4.5	13:00～17:30	-		
6/8	1	3	09:00～12:00	12:00～13:00	AWS入門	講義・演習：AWSの基本知識、コマンド演習 実施内容： Amazon Web Servicesでの設定操作を実習を通して身に付けます。 習得スキル： クラウドの基礎知識、VPCやインスタンスなどのインフラリソース構築スキル 使用プログラミング言語： なし 使用ツール： Terraterm、AmazonWebServices
		4.5	13:00～17:30	-		
6/9	1	3	09:00～12:00	12:00～13:00	AWS入門	同上
		4.5	13:00～17:30	-		
6/10	1	3	09:00～12:00	12:00～13:00	IaC入門	講義・演習：IaCの基本知識、コマンド演習 実施内容： IaCツールとして知られているTerraformを使用して、AWS上にVPCやインスタンスなどのITリソースを構築するスキルを身に付けます。AIの基礎知識を学習後、AIを使用したHCLコード生成スキルを習得します。 習得スキル： Terraformの基礎知識とコード作成スキル、AIの基礎知識とAIを使用したHCLコード生成スキル構築スキル 使用プログラミング言語： HCL 使用ツール： Terraform、Terraterm
		4.5	13:00～17:30	-		
6/11	1	3	09:00～12:00	12:00～13:00	プロジェクトマネジメント入門とAI開発運用設計	講義・演習：プロジェクトマネジメントとAI開発運用設計 実施内容： ハッカソン（チーム開発）に向けた準備として、プロジェクトマネジメントの基本知識とアジャイル手法に基づく進行管理を学びます。要件定義からのタスクの洗い出し（WBS）、タスクの可視化と進捗管理、工数、チーム内コミュニケーション、および開発中の障害に備えるリスク管理の実践的な手法を演習形式で習得します。README、要件定義書、設計書、ADR、Coding RuleをAIへ適切に与える方法を習得します。 習得スキル： プロジェクトマネジメント基礎、タスク分割（WBS）と工数見積もり、進捗管理・タスク管理能力、リスク管理
		4.5	13:00～17:30	-		
6/12	休講日					
6/13	休講日					

7/20					
6/14	1	3	09:00～12:00	12:00～13:00	【チーム演習】AI駆動開発ハッカソン① 販売・在庫・注文管理システムの要件定義とPoC検証 講義・演習：【グループワーク】対象業務の整理とDifyによる注文受付フローのPoC構築 実施内容： 販売・在庫・注文管理システムの利用者、業務上の課題、必要な機能をチームで整理します。ユーザーストーリーと受入条件を作成し、実装する機能と対象外とする機能を決定します。その後、Difyを用いて、商品と注文数量の入力、在庫確認、注文可否の判定、処理結果の表示までを行う注文受付フローのPoCを構築します。完成したPoCをチーム内で操作し、業務ルール、入力項目、エラーメッセージなどの妥当性を検証します。 習得スキル： チームでの要件定義、スコア設定、ユーザーストーリー・受入条件の作成、PoCによる仕様検証、AIを活用した要件整理 使用プログラミング言語： なし 使用ツール： Dify、設計ツール 主な成果物： 簡易要件定義書 ユーザーストーリー 受入条件 対象機能・対象外機能一覧 画面イメージ Difyによる注文受付PoC
		4.5	13:00～17:30	-	
6/15	1	3	09:00～12:00	12:00～13:00	【チーム演習】AI駆動開発ハッカソン② API・テスト設計とTDD実装開始 【グループワーク】販売・在庫・注文管理APIの設計とpytest/Jestを用いたTDD実装 実施内容： 要件とPoCの検証結果をもとに、商品・在庫・注文を扱うAPIの仕様を策定します。正常な注文、在庫不足、存在しない商品、不正な注文数量などのテストケースをAIと協働して抽出します。チーム内で商品管理、在庫管理、注文管理、画面表示などの担当を分担し、テストを先に作成してから実装するTDDのサイクルを開始します。 習得スキル： REST API設計、テストケース設計、モジュール分割、チームでの役割分担、AI駆動開発、TDDの実践 使用プログラミング言語： Python、JavaScript 使用ツール： VS Code、Git、GitHub Copilot、pytest、Jest 主な成果物： API仕様書 テストケース一覧 モジュール分担表 pytest/Jestによる自動テスト 商品・在庫・注文モジュールの初期実装
		4.5	13:00～17:30	-	
6/16	1	3	09:00～12:00	12:00～13:00	【チーム演習】AI駆動開発ハッカソン③ TDDによる販売・在庫・注文機能の実装 【グループワーク】モジュール別TDD実装とベアプログラミング 実施内容： 商品・在庫・注文の各モジュールをTDDで実装します。注文登録時の在庫確認、在庫不足時の注文拒否、注文成立時の在庫数更新など、複数のモジュールにまたがる業務処理を実装します。フロントエンドでは、商品・在庫一覧、注文入力、注文結果、注文履歴を確認できる画面を作成します。一定時間ごとに担当を交代してベアプログラミングを行い、AIが生成したコードについても、処理内容やテスト結果をチーム内で確認します。 習得スキル： 業務ロジックの実装、フロントエンドとバックエンドの連携、協調的コーディング、ベアプログラミング、TDDサイクルの継続的な実践 使用プログラミング言語： Python、JavaScript 使用ツール： VS Code、Git、GitHub Copilot、pytest、Jest 主な成果物： 商品・在庫一覧機能 注文登録機能 在庫確認・在庫更新機能 注文履歴表示機能 単体テスト モジュール間の結合処理
		4.5	13:00～17:30	-	

6/17	1	3	09:00～12:00	12:00～13:00	【チーム演習】AI駆動開発ハッカソン④ CI/CD統合・コード監査と成果発表	【グループワーク】コード統合、自動テスト、ピアレビュー、デモ発表、振り返り 実施内容: 各メンバーが実装したコードをGitHub上で統合し、GitHub Actionsを用いてpytest/Jestによる自動テストを実行します。 単体テストと結合テストの結果を確認し、不具合が発生した場合は、原因の調査、修正、再テストを行います。AIによるコードレビューとチームメンバーによるピアレビューを実施し、要件や受入条件を満たしていることを確認します。 最後に、完成した販売・在庫・注文管理システムのデモを行います。各チームが、システムの構成、担当した機能、AIの活用方法、発生した失敗やバグ、そこから得た学びを発表し、開発を振り返ります。 習得スキル: Gitを用いたコード統合、CI/CD運用、結合テスト、コードレビュー、品質保証、成果発表、開発経験の言語化 使用プログラミング言語: Python、JavaScript 使用ツール: VS Code、Git、GitHub Copilot、GitHub Actions、pytest、Jest 主な成果物: 統合済みの販売・在庫・注文管理システム GitHub Actionsによる自動テスト コードレビュー記録 テスト結果 システムデモ チーム振り返り資料
		4.5	13:00～17:30	-		
6/18	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発① 課題設定・要件定義とAPIプロトタイピング	【グループワーク】開発テーマの決定と動くモックアップの構築 実施内容: 提示されたテーマをもとに、グループごとに対象とする利用者、解決したい課題、システムが提供する価値を整理します。 ユースケースストーリーと受入条件を作成し、開発期間内に実現する機能と対象外とする機能を決定します。受入条件には、どのような入力に対して、どのような結果になれば機能が完成したと判断できるかを明記します。AIコーディングツールを活用してフロントエンドのモックアップを構築し、画面構成、画面遷移、操作方法、入力項目などをチーム内で検証します。 習得スキル: 課題設定、要件定義、スコープ設定、ユースケースストーリー・受入条件の作成、AIを用いたプロトタイピング、テスト可能な要件の定義 使用プログラミング言語: HTML、CSS、JavaScript 使用ツール: VS Code、Git、GitHub Copilot、設計ツール
		4.5	13:00～17:30	-		
6/19	休講日					
6/20						
6/21	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発② アーキテクチャ・テスト設計とスプリントプランニング	【グループワーク】システム設計、テスト設計、開発タスクの計画 実施内容: 作成した要件、受入条件、モックアップをもとに、システム構成、データベース、API、モジュール構成を設計します。 ユースケースストーリーと受入条件からテストケースを抽出し、正常系、異常系、境界値に分類します。また、各機能をTDDのサイクルで実装できる単位に分解し、優先順位、担当者、完了条件を決定します。 Gitリポジトリを作成し、ブランチの命名方法、コミットの単位、マージ方法など、チームで使用するGitの運用ルールも決定します。 習得スキル: アーキテクチャ設計、DB設計、API設計、テストケース設計、タスク分解、スプリントプランニング、Git運用設計 使用プログラミング言語: なし 使用ツール: Git、設計ツール、タスク管理ツール
		4.5	13:00～17:30	-		
6/22	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発③ スプリント実行：バックエンドのTDD実装	【グループワーク】ディリースクラムとバックエンド機能のテスト駆動開発 実施内容: ディリースクラムで進捗、当日の作業、発生している問題を共有した後、チーム内で担当を分けてバックエンドを実装します。 最初に失敗するテストを作成し、そのテストを通すために必要な最小限のコードを実装します。テスト成功後にコードを整理する一連のTDDサイクルを繰り返します。 API、データベースアクセス、入力チェック、エラー処理などを実装し、機能単位でGitのブランチとコミットを作成します。 習得スキル: ディリースクラム、バックエンド実装、データベース連携、API実装、TDD、Gitを用いたチーム開発 使用プログラミング言語: Python 使用ツール: VS Code、Git、GitHub Copilot、pytest
		4.5	13:00～17:30	-		

6/23	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発 ④ スプリント実行：フロントエンドのTDD実装	【グループワーク】UIのテスト駆動開発とバックエンドAPIの連携 実施内容: 作成したモックアップをもとに、利用者が操作するフロントエンド画面を実装します。 画面に表示する内容、利用者の操作、入力チェック、APIの呼び出し結果などについて、Jestを用いてテストを先に作成します。その後、テストを通すために必要なHTML、CSS、JavaScriptを実装する一連のTDDサイクルを繰り返します。 バックエンドAPIと連携し、データの取得、登録、更新、エラー表示までを確認します。実装は機能単位のGitブランチ上で行います。 習得スキル: フロントエンド実装、TDD、非同期通信、API連携、入力チェック、エラー表示、Gitを用いた並行開発 使用プログラミング言語: JavaScript、HTML、CSS 使用ツール: VS Code、Git、GitHub Copilot、Jest
		4.5	13:00～17:30	-		
6/24	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発 ⑤ スプリント実行：中核機能のTDD実装と改善	【グループワーク】AIを活用した業務ロジックのテスト駆動開発 実施内容: 各グループが開発するシステムの中核となる業務ロジックや、複数の条件が関係する処理をTDDで実装します。 正常系だけでなく、不正な入力、境界値、データが存在しない場合、処理途中でエラーが発生した場合などのテストを先に作成します。テストの失敗を確認してから実装し、すべてのテストが成功した後にコードをリファクタリングします。 進捗に応じてバックログの優先順位を見直し、期間内に完成させる機能をチームで調整します。 習得スキル: AIを活用した業務ロジック実装、TDD、例外処理、状態管理、境界値テスト、リファクタリング 使用プログラミング言語: Python、JavaScript 使用ツール: VS Code、Git、GitHub Copilot、pytest、Jest
		4.5	13:00～17:30	-		
6/25	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発 ⑥ システム統合と結合テスト	【グループワーク】Gitによるモジュール統合と利用シナリオに基づくテスト 実施内容: 各メンバーが開発した機能について、Gitの差分とテスト結果を確認します。確認が完了したブランチを順番にマージし、バックエンドとフロントエンドを統合します。 統合後は既存の単体テストをすべて再実行し、マージによる影響がないことを確認します。さらに、ユースケースストーリーと受入条件をもとに、利用者の一連の操作を想定した結合テストとE2Eテストを実施します。 発見した不具合については、最初に不具合を再現するテストを追加し、テストが失敗することを確認してから修正します。 習得スキル: Gitによるブランチ統合、差分確認、回帰テスト、結合テスト、E2Eテスト、テストを用いた不具合修正 使用プログラミング言語: Python、JavaScript 使用ツール: VS Code、Git、CI/CDツール、pytest、Jest
		4.5	13:00～17:30	-		
6/26	休講日					
6/27						
6/28	1	3	09:00～12:00	12:00～13:00	チームアジャイル開発 ⑦ コードレビュー・品質保証	【グループワーク】コード監査、最終テスト、スプリントレビュー 実施内容: Gitのコミット履歴と差分を利用し、AIが生成したコードを含む実装内容についてピアレビューを行います。 仕様との不整合、処理内容を説明できないコード、重複処理、入力値の検証漏れ、例外処理の不足などがないか確認します。修正が必要な場合は、対象となるテストを追加または修正した後、実装コードを変更します。 単体テスト、結合テスト、受入テストを再実行し、すべての受入条件を満たしていることを確認します。 習得スキル: コードレビュー、AI生成コードの検証、テストによる品質保証、Gitの履歴・差分の活用、成果発表 使用プログラミング言語: Python、JavaScript 使用ツール: VS Code、Git、GitHub Copilot、pytest、Jest
		4.5	13:00～17:30	-		

6/29	1	3	09:00～12:00	12:00～13:00	発表会準備	【グループワーク】最終成果発表に向けた発表資料・デモ・質疑応答の準備 実施内容: チーム開発で作成した成果物をもとに、解決を目指した課題、対象利用者、システムの概要、主要機能、システム構成、開発プロセスを整理します。 TDDの実践内容、AIの活用方法、開発中に発生した問題と解決方法を整理し、発表資料へ反映します。 システムデモの手順、説明担当、操作担当、質疑応答担当を決定し、制限時間内で発表できるようリハーサルを行います。 デモ環境、使用データ、起動手順、バックアップとなる画面資料を確認し、発表時のトラブルに備えます。 講師およびチームメンバーからのフィードバックをもとに、発表内容、資料、デモ手順を修正します。 習得スキル: 開発成果を構造化して説明するプレゼンテーション能力 システムの設計、実装、テスト、AI活用方法を他者へ説明する能力 システムデモの計画・実施能力 想定質問の整理と質疑応答への対応力 チーム内での役割分担、進行管理、最終品質確認 使用プログラミング言語: 開発したシステムで使ったプログラミング言語 使用ツール: プレゼンテーション作成ツール、開発したシステムで使った各種ツール 主な成果物: 最終成果発表資料 システムデモ手順書 発表シナリオおよび役割分担表 想定質問・回答一覧 各種設計資料 最終確認済みのシステムおよびテスト結果
		4.5	13:00～17:30	-		
6/30	1	3	09:00～12:00	12:00～13:00	最終成果発表・決意表明&クロージング	【個人ワーク・グループワーク】最終成果発表、研修全体の振り返り、決意表明 実施内容: 各チームが開発したシステムについて、課題、システム概要、主要機能、設計上の工夫、TDDの実践内容、AIの活用方法、発生した問題と解決方法を発表します。 システムのデモを行い、要件や受入条件を満たしていること、主要機能が正しく動作することを説明します。 発表後の質疑応答では、システム全体の構成、コード、テスト、AI生成物の検証方法について回答します。 講師および他チームからのフィードバックを受け、良かった点、改善すべき点、今後取り組む課題を整理します。 研修全体を振り返り、配属後にAIをどのように活用し、開発成果への説明責任をどのように果たすかを個人で言語化し、決意表明を行います。 研修で得た学び、今後の学習方針、現場で実践する行動を共有し、研修を終了します。 習得スキル: 開発成果を論理的かつ簡潔に説明する能力 システムデモおよび質疑応答への対応力 AI生成コードを含む開発成果に対する説明責任 フィードバックを受けて改善点を整理する振り返り能力 学習成果と今後の行動目標を言語化する能力 使用プログラミング言語: 開発したシステムで使ったプログラミング言語 使用ツール: プレゼンテーション作成ツール、開発したシステムで使った各種ツール 主な成果物: 最終成果発表資料 完成したシステムおよびデモ 質疑応答記録およびフィードバック記録 個人振り返り 配属後のAI活用方針および決意表明
		4.5	13:00～17:30	-		
合計	22	165				