

科目名	プログラミング 2							年度	2024
英語科目名	Programming 2							学期	後期
学科・学年	ゲームクリエイター科四年制 ゲームプログラマーコース 1年次	必／選	必	時間数	60	単位数	4	種別※	講義+演習
担当教員	鈴木靖生	教員の実務経験		有	実務経験の職種		プログラマー		
【科目の目的】 ・C++言語を使用して複雑なプログラムを記述できる。 ・クラスとその組み合わせ方を理解できる。 ・C++言語のライブラリを理解できる。									
【科目の概要】 ゲームプログラマの必須スキルである、C++言語の文法を講義と演習を通じての理解を目指します。 また、オブジェクト指向プログラミングの考え方やアルゴリズムについても学びます。									
【到達目標】 A. C++言語の文法を理解している。 B. クラスの組み合わせ方を理解している。 C. C++言語の標準ライブラリを理解している。 D. C++言語のコンテナについて理解している。									
【授業の注意点】 配布資料をよく理解し、予習・復習を行うこと。 なお、テストやパフォーマンス課題は欠席・未提出となると0点になるため注意すること。									
評価基準＝ルーブリック									
ルーブリック 評価	レベル5 優れている	レベル 4 よい		レベル3 ふつう		レベル2 あと少し		レベル1 要努力	
到達目標 A	文法に則って、課題の 8割を完成できる。			文法に則って、課題の 半分の完成できる。				文法に則ってプログラ ムを記述できない。	
到達目標 B	クラスを組み合わせ て、課題の8割を完成 できる。			クラスを組み合わせ て、課題の完成でき る。				クラスを組み合わせか たを知らない。	
到達目標 C	ライブラリを利用し て、課題の8割を完成 できる。			ライブラリを利用し て、課題の半分の完成 できる。				ライブラリの詳細を知 らない。	
到達目標 D	コンテナを選択して、 課題の8割を完成でき る。			コンテナを選択して、 課題の半分の完成でき る。				コンテナの種類と特性 を知らない。	
【教科書】 特になし									
【参考資料】 クラスの基本C++/PDF、参考になるサイトを授業中に紹介する。									
【成績の評価方法・評価基準】 授業中に実施するテスト、プログラムの提出、Webテスト等で、総合的に評価する。									
※種別は講義、実習、演習のいずれかを記入。									

科目名		プログラミング 2				年度	2024
英語表記		Programming 2				学期	後期
回数	授業テーマ	各授業の目的	授業内容		到達目標＝修得するスキル	評価方法	自己評価
1	クラスの基礎	クラスをどのように作成し実行するのか	1	クラスの作成	クラスの書き方を理解している	2	
			2	クラスの実行	クラスの実行方法を理解している		
2	クラスのファイル分割	クラスをどのようにファイル分割するのか	1	ファイルの分割	ファイルの分割方法を理解している	2	
3	クラスの関連	クラスをどのように関連させるのか	1	クラスの集約	クラスの集約方法を理解している	2	
			2	クラスの継承	クラスの継承方法を理解している		
4	デザインパターン	デザインパターンをどのように活用するのか	1	集約と継承 1	テンプレートメソッドを理解している	2	
			2	集約と継承 2	ストラテジーパターンを理解している		
5	標準入力、標準出力	標準入力、標準出力をどのように使うのか	1	コンソール	コンソールへの入出力を理解している	2	
			2	ファイル	ファイルへの入出力を理解している		
6	コンテナ 1	コンテナをどのように使うのか	1	コンテナ	コンテナ使用の意義を理解している	2	
			2	vector	vectorの特性を理解している		
7	コンテナ 2	コンテナをどのように使うのか	1	list	listの特性を理解している	2	
			2	map	mapの特性を理解している		
8	クラスのコピー	クラスをどのようにコピーするのか	1	クラスのコピー	クラスのコピー方法を理解している	2	
			2	コピーの禁止	クラスのコピー禁止方法を理解している		
9	テンプレート	テンプレートをどのように使うのか	1	テンプレート	テンプレートの意義を理解している	2	
			2	テンプレート	テンプレートの作成方法を理解している		
10	雑則	その他、文法を紹介する	1	const	constの使用方法を理解している	2	
			2	参照	値の参照方法を理解している		
11	複雑なプログラムの改良 1	どのようにプログラムを改良するのか	1	企画	企画の立て方を理解している	2	
			2	仕様書	仕様書の書き方を理解している		
12	複雑なプログラムの改良 2	どのようにプログラムを改良するのか	1	設計	設計の考え方を理解している	2	
			2	クラス図	クラス図の書き方を理解している		
13	複雑なプログラムの改良 3	どのようにプログラムを改良するのか	1	コーディング	C++言語の文法を理解している	2	
			2	テスト	テスト方法を理解している		
			3	デバック	バグを取る方法を理解している		
14	複雑なプログラムの改良 4	どのようにプログラムを改良するのか	1	コーディング	C++言語の文法を理解している	2	
			2	テスト	テスト方法を理解している		
			3	デバック	バグを取る方法を理解している		
15	複雑なプログラムの改良 4	どのようにプログラムを改良するのか	1	コーディング	C++言語の文法を理解している	2	
			2	テスト	テスト方法を理解している		
			3	デバック	バグを取る方法を理解している		

評価方法：1. 小テスト、2. パフォーマンス評価、3. その他

自己評価：S：とてもよくできた、A：よくできた、B：できた、C：少しできなかった、D：まったくできなかった

備考 等