

学期 / Semester	2021年度 / Academic Year 後期 / Second Semester	曜日・校時 / Day・Period	月 / Mon 4, 月 / Mon 5
開講期間 / Course duration	2021/09/28 ~ 2022/03/31		
必修選択 / Required / Elective	必修 / required	単位数(一般/編入/留学) / Credits (General / Transfer/Overseas)	2.0/2.0
時間割コード / Time schedule code	20213802005001	科目番号 / Course code	38020050
科目ナンバリングコード / Numbering code	ID-ID-1-050-1-112		
授業科目名 / Course title	プログラミング演習 / Practice in Software Programming		
編集担当教員 / Instructor in charge of the course syllabus	松永 昭一 / Matsunaga Shoichi		
授業担当教員名 (科目責任者) / Instructor in charge of the course	松永 昭一 / Matsunaga Shoichi		
授業担当教員名 (オムニバス科目等) / Instructor(s)	松永 昭一 / Matsunaga Shoichi		
科目分類 / Course Category	共通科目, 情報学基盤科目 (コンピュータ科学)		
対象年次 / Intended year	1	講義形態 / Course style	演習 / Seminar
教室 / Class room	[総合]総合教育研究棟2F多目的ホール, [情報] 1号館 4 F 情報システム工学科演習室		
対象学生 (クラス等) / Intended year (class)	情報データ科学部1年次		
担当教員Eメールアドレス/E-mail address	mat@cis.nagasaki-u.ac.jp		
担当教員研究室/Office	情報データ科学部1号館2階 情報システム研究室205		
担当教員TEL/Tel	2700		
担当教員オフィスアワー/Office hours	月5校時. これ以外でも良いがアポイントを奨めます.		
授業の概要及び位置づけ/Course overview	情報系の学生が習得すべき基礎能力であるプログラミングの基礎概念に習熟する。同時に、プログラミング環境の設定方法を学ぶ		
授業到達目標/Course goals	プログラミング言語pythonの基本制御構造や構文を理解し、与えられたプログラムを読んで意味を理解できる。論理的な制御構造を用いて、与えられた仕様に基づく、コメントや関数使用によって可読性を考慮したプログラムを自力で設計・作成できる。		
知識・技能以外に、この授業を通して身につけて欲しい力 (1つ以上3つまで) / Abilities other than knowledge and skills acquired mainly through the course ( pick 1 to 3 )	主体性 / Autonomy 汎用的能力 / Generic Competence 倫理観 / Ethics 多様性の理解 / Understanding Diversity 協働性 / Cooperativeness 考えをやり取りする力 / Ability to exchange ideas 国際・地域社会への関心 / Interest in international / local society		
学生の思考を活性化させるための授業手法/Teaching method to stimulate students' thinking	A. 授業内容の理解度を確認したり自分で考えさせたりする活動   / Activities to check the degree of comprehension of the contents to the lesson or to think over B. 多角的に考えるために他者と関わる活動   / Activities involving others to think from various perspectives C. 技能修得のために実践する活動   / Activities to practice for acquiring skills D. 問題解決のために知識を総合的に活用する活動   / Activities that comprehensively utilize knowledge to solve problems E. 上記以外の学生の思考の活性化を促す授業手法   / Teaching methods to stimulate students' thinking other than the above F. 教員からの講義のみで構成される   / It consists only of lectures from teachers		
成績評価の方法・基準等/Method of evaluation	講義時間中に取り組み課題の達成度およびレポート課題の内容を評価し、その評価点が60%以上のものを合格とする。ただし、全回出席し、全てのレポートを提出することを前提とする。		
各回の授業内容・授業方法 (学習指導方法) /Course contents of each lesson	詳細は授業計画詳細を参照		
事前、事後学習の内容/Preparation & Review	授業計画に沿った事前、事後学習を行うこと。演習内容を効率良く理解するため、そして演習時になされる教員の質問に解答することができるためにも、その回に取り上げられるトピックを予習してることが推奨される。(1h) 各回の演習トピックは多岐にわたるとともに、相互に関連している。各演習の直前に復習したのでは、正確な理解を身につけることは難しいため、毎回復習すること(1h)		
キーワード/Keywords	プログラミング, 演算, 条件分岐, 繰り返し構造, 入出力, 配列		
教科書・教材・参考書/Materials	プログラミング概論と同じ教科書を使用する。 参考書 Bill Lubanovic 著 (監訳: 斎藤康毅, 訳: 長尾高弘), 入門Python3, オライリー・ジャパン, オー		
受講要件 (履修条件) /Prerequisites	原則として全回出席を前提とする。ただし、やむを得ず正当な理由で欠席する場合は個別指導を行うので担当教員に連絡すること。		

アクセシビリティ/Accessibility (for students with disabilities)	長崎大学では、全ての学生が平等に教育を受ける機会を確保するため、修学の妨げとなり得る社会的障壁の除去及び合理的配慮の提供に取り組んでいます。授業における合理的配慮等のサポートについては、担当教員（上記連絡先参照）または「アシスト広場」（障がい学生支援室）にご相談下さい。 アシスト広場（障がい学生支援室）連絡先 (TEL) 095-819-2006 (FAX) 095-819-2948 (E-MAIL) support@m1.nagasaki-u.ac.jp
備考 (URL) /Remarks (URL)	lacsのプログラミング演習Iに記載する
学生へのメッセージ/Message for students	受講時には各自のノートパソコンを持参すること。やむを得ず欠席する場合は個別指導を行うので担当教員に連絡すること。
実務経験のある教員による授業科目であるか (Y/N)/Instructor(s) with practical experience	Y
実務家教員名 / 実務経験内容 / 実務経験に基づく教育内容 (実務経験のある教員による授業科目のみ使用) /Name / Details of practical experience / Contents of course	松永昭一 / 日本電信電話において研究開発に従事 / プログラミングを実務上どのように用いていたかを例示し、学生の理解度を深める
授業計画詳細 / Course Schedule	
回(日時) / Time(date and time)	授業内容 / Contents
第1回	オリエンテーション ~ 全回の授業計画についての説明、課題提出のためのLACS、及び教職員やティーチングアシスタント(TA)との授業中のコミュニケーションツールについての説明
第2回	演習準備1 (プログラミングに必要なツール類とその役割を理解する) ~ google chrome, emacs, python 3.8.6 のインストールができる, マイクロソフトWindows10のGUI(Graphical User Interface)が使える
第3回	演習準備2 (プログラミングに必要なツール類を使うことができる) ~ マイクロソフトWindows10のCUI(Character User Interface) (コマンドプロンプトの設定、cd, dir, ren, copy, md, delなどの各種コマンド演習)、pythonの対話モードの使い方、emacsでの簡単なプログラムの作成実行、emacsの操作方法、pythonのバッチモードでの実行
第4回	型と演算 (目的に応じた型が理解できる) ~ コメント, 整数, 実数, リテラル, 算術演算子, 変数, 型変換
第5回	基本的な入出力と文字列の扱い (基本的な入出力を制御したプログラムを作成できるとともに文字列の結合ができる) ~ 文字列リテラル (エスケープ文字含む), 連結+と繰り返し*, len, index指定, スライス指定, input
第6回	演算子 (型に応じた演算子を使用できる) ~ 比較演算子, 論理演算子
第7回	条件分岐 (if文が使える。インデントの意味を理解し使うことができる) ~ if, elif, else:とインデントの使い方
第8回	例外処理と描画入門 (例外処理と簡単な描画ができる) ~ try, except, matplotlib
第9回	データセット (データを格納・操作できる) ~ リスト[ ], タプル( )
第10回	制御構造の多重化 (制御構造を多重化したプログラムを作成できる) ~ if文, for文, while文を高度に利用
第11回	データセット (高度なデータ格納・操作できる) ~ del文, コンテナ, in 演算子, 辞書{ }, 集合{ }, ソート, unpacking, 内包表記
第12回	関数 (関数の定義と利用ができる) ~ 位置引数, キーワード引数, ローカル変数, 大域変数, ラムダ式
第13回	関数 (高度な関数の定義と利用ができる) ~ 可変長位置引数, 可変長キーワード引数, ライブラリの自作, 再帰的定義
第14回	オブジェクト指向とクラス (オブジェクト指向のプログラミングを理解する) ~ オブジェクト, クラス, インスタンス
第15回	最終課題 (学習したことを総合的に用いたプログラムを作成する) ~ google colabを利用する
第16回	最終課題の評価と指導

学期 / Semester	2021年度 / Academic Year 前期 / First Semester	曜日・校時 / Day・Period	水 / Wed 3, 水 / Wed 4
開講期間 / Course duration	2021/04/01 ~ 2021/09/27		
必修選択 / Required / Elective	必修 / required	単位数(一般/編入/留学) / Credits (General / Transfer/Overseas)	2.0/2.0
時間割コード / Time schedule code	20213802005501	科目番号 / Course code	38020055
科目ナンバリングコード / Numbering code	ID-ID-1-055-1-112		
授業科目名 / Course title	プログラミング演習 / Practice in Software Programming II		
編集担当教員 / Instructor in charge of the course syllabus	藺田 光太郎 / Sonoda Kotaro		
授業担当教員名 (科目責任者) / Instructor in charge of the course	藺田 光太郎 / Sonoda Kotaro		
授業担当教員名 (オムニバス科目等) / Instructor(s)	藺田 光太郎 / Sonoda Kotaro		
科目分類 / Course Category	共通科目, 情報学基盤科目 (コンピュータ科学)		
対象年次 / Intended year	2	講義形態 / Course style	演習 / Seminar
教室 / Class room	[総合]総合教育研究棟2F多目的ホール		
対象学生 (クラス等) / Intended year (class)	2年次		
担当教員Eメールアドレス/E-mail address	sonoda-2021ps2@cis.nagasaki-u.ac.jp		
担当教員研究室/Office			
担当教員TEL/Tel	095-819-2586		
担当教員オフィスアワー/Office hours			
授業の概要及び位置づけ/Course overview	代表的なプログラミング言語であるPythonを用いて, 計算機科学や情報工学を専攻する者にとって必須なプログラミング実装能力・プログラミング読解力, デバッグ能力, および可読性の高いプログラム作成能力を習得する.		
授業到達目標/Course goals	モジュール, 自作関数を駆使し, オブジェクト指向に基づくプログラムを簡潔に可読性高く作成できる.		
知識・技能以外に, この授業を通して身につけて欲しい力 (1つ以上3つまで) / Abilities other than knowledge and skills acquired mainly through the course (pick 1 to 3)	主体性 / Autonomy 汎用的能力 / Generic Competence 倫理観 / Ethics 多様性の理解 / Understanding Diversity 協働性 / Cooperativeness 考えをやり取りする力 / Ability to exchange ideas 国際・地域社会への関心 / Interest in international / local society		
学生の思考を活性化させるための授業手法/Teaching method to stimulate students' thinking	A. 授業内容の理解度を確認したり自分で考えさせたりする活動   / Activities to check the degree of comprehension of the contents to the lesson or to think over B. 多角的に考えるために他者と関わる活動   / Activities involving others to think from various perspectives C. 技能修得のために実践する活動   / Activities to practice for acquiring skills D. 問題解決のために知識を総合的に活用する活動   / Activities that comprehensively utilize knowledge to solve problems E. 上記以外の学生の思考の活性化を促す授業手法   / Teaching methods to stimulate students' thinking other than the above F. 教員からの講義のみで構成される   / It consists only of lectures from teachers		
成績評価の方法・基準等/Method of evaluation	作成したプログラムソースコードと, その実行結果をレポートとして毎回提出. すべての課題に対してレポートを提出し, その評価点が60%以上のものを合格とする. 全回出席を前提とするが, やむを得ず欠席の場合は, 個別指導をおこなうので担当教官に連絡すること.		
各回の授業内容・授業方法 (学習指導方法) /Course contents of each lesson	詳細は授業計画詳細を参照		
事前・事後学習の内容/Preparation & Review	授業で指示する.		
キーワード/Keywords	Python, プログラミング, 関数, モジュール, クラス, オブジェクト指向, データサイエンス		
教科書・教材・参考書/Materials	参考書: コーリー・アルソフ, 清水川貴之 (監訳) 「独学プログラマー Python言語の基本から仕事のやり方まで」 教科書は特に指定しない. 教員のホームページで課題を出題.		
受講要件 (履修条件) /Prerequisites	プログラミング演習 を履修したものととして演習をすすめる		
アクセシビリティ/Accessibility (for students with disabilities)	長崎大学では, 全ての学生が平等に教育を受ける機会を確保するため, 修学の妨げとなり得る社会的障壁の除去及び合理的配慮の提供に取り組んでいます. 授業における合理的配慮等のサポートについては, 担当教員 (上記連絡先参照) または「アシスト広場」 (障がい学生支援室) にご相談下さい。 アシスト広場 (障がい学生支援室) 連絡先 (TEL) 095-819-2006 (FAX) 095-819-2948 (E-MAIL) support@ml.nagasaki-u.ac.jp		

備考 (URL) /Remarks (URL)	
学生へのメッセージ/Message for students	予習・復習をしっかりと取り組むこと。
実務経験のある教員による授業科目であるか (Y/N)/Instructor(s) with practical experience	
実務家教員名 / 実務経験内容 / 実務経験に基づく教育内容 (実務経験のある教員による授業科目のみ使用) /Name / Details of practical experience / Contents of course	
授業計画詳細 / Course Schedule	
回(日時) / Time(date and time)	授業内容 / Contents
授業方法	その日のポイントするプログラム表現手法に応じて課す課題の演習を中心とする。
第 1 回	プログラミング演習 の復習 ( 1 )
第 2 回	プログラミング演習 の復習 ( 2 )
第 3 , 4 回	関数, モジュール
第 5 , 6 , 7 回	クラス, インスタンス, 継承
第 8 , 9 回	ファイル入出力と多次元配列に対する処理
第10 , 1 1 回	数値計算
第 1 2 回	可視化
第 1 3 , 1 4 , 1 5 回	実践課題
第 1 6 回	まとめ