

実践論文

コロナ禍におけるプログラミング演習科目の オンライン授業実践とその考察

岡本 雅子^{1,a)}

受付日 2021年9月30日, 再受付日 2022年1月10日,
採録日 2022年2月19日

概要: 京都大学では、2020年3月、コロナ禍における対策として、ほとんどすべての対面授業をオンライン受講できるよう対策するとの方針を示した。本稿では、こうした状況下において実施した演習型プログラミング授業について、その実践例を報告するとともに、2019年に実施した従来型の対面授業と比較し、演習型授業のオンライン化における課題や問題点などについて考察した。その結果、オンライン授業では、授業の前半でつまずく受講者が多く見られたが、授業の後半では課題のプログラムを完成することができた割合が高かった。オンライン授業では、受講者が授業時間内にチャットで積極的に質問して、個々の疑問点を解消していたことが推察された。一方、対面授業では、授業の後半でつまずく受講者が多く見られており、授業中に質問しなかった（できなかった）受講生がいた可能性が浮かび上がった。

キーワード：プログラミング, 演習, オンライン授業, 初学者, コロナ禍

A Practice and Consideration of Online Class for Computer Programming in COVID-19

MASAKO OKAMOTO^{1,a)}

Received: September 30, 2021, Revised: January 10, 2022,
Accepted: February 19, 2022

Abstract: Due to the need to prevent the spread of the COVID-19 in March 2020, Kyoto University has been requested to offer online classes. Therefore, it has become necessary to shift from face-to-face classes to online classes. In this paper, a practical example of programming class is reported. Additionally, in this study, problems and issues in online classes are discussed by comparing with previous face-to-face classes in 2019. As a result, in the online class, many students failed in the first half of the class, but a high percentage of them were able to complete the assigned program in the second half of the class. It was assumed that the students actively asked questions via chat during the class session to resolve their individual issues. On the other hand, in the face-to-face class, many students appeared to falter in the latter half of the class, suggesting the possibility that some students did not (or could not) ask questions during the class.

Keywords: programming, practice class, online education, novices, COVID-19

1. はじめに

京都大学は、ICTや図書館利用のスキルを扱う演習科目として、前期に文科系学部生を対象とした「情報基礎演習(全学向け)」を開講している [1]。本科目は、教養教育に

おけるコンピュータリテラシ教育という位置付けで、コンピュータやネットワークといった教育・学習におけるICT利用に関する基礎知識を獲得するとともに、学術情報の探索、学術文書の作成、表計算ソフトを用いた分析やシミュレーション、プレゼンテーション、プログラミングについて演習形式で学習することになっている [2], [3]。これまで、同科目は、教育用コンピュータシステムのPC端末が設置されている演習室において対面で演習形式の授業を実施し

¹ 京都大学高等教育研究開発推進センター
Kyoto University, Kyoto 606-8501, Japan

^{a)} okamoto.masako.8v@kyoto-u.ac.jp

てきた [4], [5]。しかしながら、2020 年 3 月に新型コロナウイルス感染症拡大防止のため、京都大学では、オンラインでの授業の実施が要請されたことにもない、2020 年度の前期の時点で、授業のオンライン化を試みることとなった。

このような現状の下、2020 年度前期は、前年度までに利用していた学内 LMS (PandA) に加え、ビデオ会議サービス (Zoom) を利用した同時双方向型のオンライン授業として実施した。さらに、2021 年度のクラスは、原則対面授業としながら、オンライン参加も認めるハイフレックス型で授業を実施することになった。

本稿では、「情報基礎演習 (全学向け)」の中でもプログラミングの授業に焦点を当て、対面形式、完全オンライン形式、およびハイフレックス型 [6] で実施したプログラミング演習について、報告するとともに、受講者の成績や学習動向などから本授業のオンライン化に際した課題や問題点などについて考察する。

2. 授業実践

2.1 対象とした授業

本稿の対象とした授業は、2019 年度から 2021 年度の前期に京都大学で実施された全学共通科目 (1 回生から 4 回生まで受講可)「情報基礎演習」である。表 1 のクラスを対象とした。2019 年度および 2020 年度は、月 4、月 5、木 2 と 3 クラスずつ開講されているが、2021 年度は時間割が変更され、木 2 のみ開講することになった。なお、各年度すべて同じ教材を使用して、同じ授業を実施している。これらのクラスは著者を含む 2 名の教員が共同で担当し、受講者がつまづいた場合には、教員やティーチングアシスタント (TA) 1 名がサポートに入り直接対応した。2019 年度の受講者数は 81 名、2020 年度は 130 名、2021 年度は 30 名であった。

本科目は、教養教育におけるコンピュータリテラシー教育という位置付けで週 1 回 90 分の選択科目として実施したもので、「アカデミックな活動に必要な ICT スキルを獲得すること」、「自立した ICT ユーザとなること」、「ICT スキルを自主的・継続的に獲得する自学自習能力を身につけること」の 3 点を目的としている。コンピュータやネットワークといった教育・学習における ICT 利用の基礎知識を獲得するとともに、学術情報の探索、学術文書の作成、表計算ソフトを用いた分析やシミュレーション、プレゼン

テーション、プログラミングについては演習形式で実施している。プログラミングの授業は、後半 4 コマの授業を割り当てた。このような科目構成から、今回は 4 コマのみでプログラミングの授業を実施しているため、プログラミングのみを半期 14 コマで学ぶ授業に比べ、相対的に時間数が少ない。

前述のように、本科目は 2019 年度までは学内の PC 教室で実施してきたが、2020 年度のクラスは感染症拡大防止のための対応として、前年度までに利用していた PandA に加え、Zoom を利用した同時双方向型のオンライン授業として実施した。さらに、2021 年度のクラスは、原則として対面授業としながら、各受講者の置かれた状況に合わせてオンライン参加も認めるハイフレックス型で授業を実施した。

2.2 授業デザインと授業形態

対象とした科目の中でプログラミングを学ぶ機会は、先述のように 1 コマ 90 分の授業が計 4 コマしかなく、学習する時間が限られていることから、学習内容は、前半 3 コマの授業でプログラミングの基本的な要素である変数と制御構造などを学習し、最終の第 4 回で 50 行から 100 行程度で作成できるプログラムを作成した (表 2 参照)。

なお、表 2 に示すように、対面で授業を実施した 2019 年度と完全オンラインおよびハイフレックス型で授業を実施した 2020・2021 年度では、「ランダムな値の生成」の学習回が異なる。

第 4 回 (授業では第 14 回) の授業は、「数当てゲーム」の作成を題材とした。90 分の授業構成は、以下のとおりである。

- (1) 数当てゲームの exe ファイルを配布し、各自が動作を確認する。
- (2) 動作確認後、アルゴリズムを考えて、フローチャートなどの図あるいは文章、矢印などで可視化する。
- (3) Python のプログラムを作成する際に必要となるコードなどの要素の確認。
- (4) ランダムな値の生成方法や入力された値のチェックなど数当てゲームのアルゴリズムの確認。
- (5) プログラムの作成。
- (6) プログラムの動作確認と修正。

また、本授業では、講義とテキスト教材による個別学習を取り入れた授業形態を採用し、講義では短時間 (30 分程度) にポイントを絞って解説を行い、残りの時間 (60 分程度) を演習の時間に割り当てた。なお、本授業では、独自に開発したテキスト教材を使用した (図 1 参照)。テキスト教材では、受講者が自立的に作業を遂行することを考慮して、サンプルプログラムは「完全に動作するプログラム」と「その実行結果」を掲載する方針をとった。そして、講義では、サンプルプログラムを対象に、プログラムの各行

表 1 対象授業の履修者数

Table 1 The number of students for each target class.

	月 4	月 5	木 2	合計
2019 年	18	20	43	81
2020 年	42	33	55	130
2021 年	-	-	30	30

表 2 プログラミングの授業内容

Table 2 Contents of computer programming class.

	授業内容 (2019 年度)	授業内容 (2020/2021 年度)
第 1 回 (授業では第 11 回)	<ul style="list-style-type: none"> 開発環境の基本操作 Python の基礎 (題材: 変数の利用, プログラムの動作する順番など) 	<ul style="list-style-type: none"> 開発環境の基本操作 Python の基礎 (題材: 変数の利用, プログラムの動作する順番など)
第 2 回 (授業では第 12 回)	<ul style="list-style-type: none"> 条件を判断して処理を実行する (題材: テストの点数の判定など) 例外処理 (題材: 入力された数値の判定) 	<ul style="list-style-type: none"> 条件を判断して処理を実行する (題材: テストの点数の判定など) 例外処理 (題材: 入力された数値の判定) ランダムな値の生成 (題材: コンピュータとのじゃんけんゲーム)
第 3 回 (授業では第 13 回)	<ul style="list-style-type: none"> 同じ処理を指定した回数だけ繰り返す (題材: Fizz Buzz 問題など) 条件が満たされている間に処理を繰り返す (題材: Fizz Buzz 問題など) ランダムな値の生成 (題材: コンピュータとのじゃんけんゲーム) 	<ul style="list-style-type: none"> 同じ処理を指定した回数だけ繰り返す (題材: Fizz Buzz 問題など) 条件が満たされている間に処理を繰り返す (題材: Fizz Buzz 問題など)
第 4 回 (授業では第 14 回)	<ul style="list-style-type: none"> 数当てゲームの作成 	<ul style="list-style-type: none"> 数当てゲームの作成

に対応して逐次解説を加えたほか、プログラムの構成要素の紹介も併せて行った。テキスト教材で使用したサンプルプログラムを図 A.1 から図 A.6 に示す。

講義については、対面授業では、受講者の様子を見ながら説明のスピードや内容を調整することができるが、オンライン授業では受講者の様子を把握することが困難なため、重要な項目や開発環境の操作方法については、講義の中で 3 回ほど繰り返して説明するようにし、対面のときよりも丁寧にゆっくり話すように意識的に実施した。加えて、オンライン授業では、テキストのページ数や見出しを適宜伝えるようにし、受講者が説明している箇所を把握しやすいように配慮した。

このほか、2019 年度の対面授業では、授業の開始時に印刷したテキスト教材を配布し、PDF 版のテキスト教材を PandA にも掲載した。2020 年度の完全オンライン型の授業では、授業の前日の午前中までに PDF 版のテキスト教材を PandA に掲載し、必要な受講者は各自が印刷するように依頼した。2021 年度のハイフレックス型の授業では、2020 年度と同様に授業の前日の午前中までに PDF 版のテキスト教材を PandA に掲載するとともに、対面の参加者には、授業の開始時に印刷したテキスト教材を配布した。

2.3 受講者の構成

本実践で対象とした受講者は、プログラミングの第 1 回 (授業では第 11 回) から第 4 回 (授業では第 14 回) の授業に出席し、課題を提出した 212 名である。2019 年度、2020 年度、2021 年度の対象者を表 3 に示す。なお、括弧内は、プログラミング経験者の人数である。

また、本授業はプログラミング初学者を想定したものであるが、経験者の履修も可能であり、本実践では、Java など他のプログラミング言語の既修者 (高等学校などでプロ

表 3 本実践の対象者

Table 3 The target students of this practice.

	月 4	月 5	木 2	合計
2019 年	38 (0)	17 (1)	16 (1)	71 (2)
2020 年	51 (3)	38 (5)	26 (3)	115 (11)
2021 年	—	—	26 (3)	26 (3)

グラミングを学習した初級者) が 16 名であった。

次に、ハイフレックス型で実施した 2021 年度の対面とオンラインの参加状況を表 4 に示す。同年度の 1 回目 (授業では第 11 回) の授業実施の時点では、オンライン授業が推奨されていたため、対面の参加者はいなかったが、第 2 回以降の授業については、受講者側の都合に合わせてオンラインでの授業参加を許可したことから、同形式での参加者が見られた。なお、オンライン参加者の内訳は、第 2 回 (授業では第 12 回) が 15 名、第 3 回 (授業では第 13 回) が 10 名、第 4 回 (授業では第 14 回) が 10 名であった。

2.4 オンラインおよびハイフレックス型授業への移行について

2020 年度の京都大学の全学共通科目については、前期の授業は 5 月 6 日まで休講とし、原則としてオンラインで実施するという方針が出された。その際、学習プラットフォームとして、本学の LMS を活用し、

- (1) Zoom を利用した同時双方向型授業
- (2) 教材配布と動画での解説によるオンデマンド型授業
- (3) 教材配布と音声またはテキストの解説によるオンデマンド型授業

という、実施方法が提示された。

5 月 7 日以降は、いずれのクラスも Zoom を利用した同時双方向型授業を実施しただけではなく、通信環境などの

変数

- ◆ 変数の利用
 - * 【演習9】以下のようにプログラムを入力し、実行してください。

```
a = 100
b = 200
c = a + b
print(c)
300
```

- ◆ 変数
 - ◆ **a = 100**
 - 計算結果や文字を格納する“a”という名前の箱です → 「**変数**」と呼ばれます
 - ◆ **a = 100** は変数 a に 100 を格納します
 - ◆ **変数 a**
- * 構文
 - ◆ 変数に値を代入するときは、代入演算子「=」を使います（数学の等式とは異なりますので注意してください！）
 - ◆ 右辺から左辺に代入される
 - ◆ **変数名 = 格納する値**
- * 【応用編】変数名のルール
 - ◆ 変数名には、英数字、「_」（アンダースコア）を使用できる
 - ◆ 最初の文字に数字は使えない
 - ◆ 変数名の大文字と小文字は区別される
 - ◆ 予約語は使えない（プログラムの命令など）

- ◆ Print文
 - * 変数の値や文字列を出力する際に用いる命令
 - ◆ **print(値)**
 - 値には変数を指定することができる
 - ◆ プログラムの動作する順番
 - * 以下のプログラムを入力し、実行してください

```
a = 2
a = 3
print('a の値は', a, 'です')
```

a の値は 3 です

- ◆ **print**文では、「」で囲まれた部分はそのまま出力される
- ◆ 「」で文節をつなげることができる
- ◆ カマドの後は、半角スペースを1つ入れてください！
- ◆ 変数に格納した値は、どんどん上書きされていきます。そのため、元の値は消えてしまいますので、注意してください！

- * print文で変数や文字列を連結する場合
 - ◆ **print('文字列', 変数名, '文字列')** のように記述することができる

プログラムの基本：
プログラムは**上から下へ**と実行される！

17 18

同じ処理を指定した回数だけ繰り返す (for文)

- ◆ 【演習1】以下のプログラムを入力し、実行してください
- ◆ 特に、インデントと処理の関係に注目してください

```
for i in range(1, 10):
    if i % 3 == 0 and i % 5 == 0:
        print('Fizz Buzz')
    elif i % 3 == 0:
        print('Fizz')
    elif i % 5 == 0:
        print('Buzz')
    else:
        print(i)
print('ここはfor文の中です')
print('ここはfor文の外です')
```

- * 【復習】インデントは「**タブ**」 or 「**スペース**」
- ◆ インデントは、Tabキーを押してスペースを入れる
- ◆ 通常、for文やif文を入力した際、「Enter」キーを押下すると、インデントした状態で改行される

- ◆ 【演習1の実行結果】

```
1
ここはfor文の中です
2
ここはfor文の中です
Fizz
3
ここはfor文の中です
Buzz
4
ここはfor文の中です
Fizz
5
ここはfor文の中です
7
ここはfor文の中です
8
ここはfor文の中です
9
ここはfor文の中です
Fizz
ここはfor文の外です
```

- ◆ **for文による繰り返し処理(1)**
- ◆ **for 変数名 in range(開始する値, 終了する値):**
- ◆ **繰り返し処理**
- ◆ 「終了する値未満になることに注意してください」

- * 「1, 2, 3, 4, 5, 6, 7, 8, 9, 10」のように10回繰り返したい場合は、
- ◆ for i in range(1, 10 + 1):
- ◆ あるいは、for i in range(1, 11): と指定する必要があります

条件が満たされている間、処理を繰り返す (while文)

- ◆ 【演習7】以下のプログラムを入力し、実行してください

```
c = 1
while c < 10:
    if c % 3 == 0 and c % 5 == 0:
        print('Fizz Buzz')
        break
    elif c % 3 == 0:
        print('Fizz')
    elif c % 5 == 0:
        print('Buzz')
    else:
        print(c)
    c = c + 1
```

- ◆ **while文の処理**
- ◆ **if文の処理**
- ◆ **条件が満たされている間、処理を繰り返し実行できる**

- ◆ 【演習7の実行結果】

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
```

- ◆ **while文による繰り返し処理**
- ◆ **while 条件式:**
- ◆ **(条件式が成り立っている間) 繰り返し処理**
- ◆ 条件式の書き方はif文と同じ
- ◆ For文と類似に「continue文」は「break文」が使用できます
- ◆ For文と類似に「continue文」は「break文」が使用できます

9 13

図 1 使用したテキスト教材（一部抜粋）

Fig. 1 Learning material for computer programming (Excerpt from the textbook).

事情に配慮し、授業時間内に Zoom に接続できなかった受講者への対応として、録画した授業を授業後に視聴可能とすることが推奨されていたことから、本科目でも PandA を通じてアーカイブ動画を受講者に提供した。

次に、2021 年前期の授業は、原則として対面授業を実施するという方針が出されたが、希望者については、オンライン参加を認めるよう配慮が求められた。そのため、本科目では、対面の受講者は、PC 教室で教育用コンピュー

表 4 2021 年度の対面・オンラインの出席状況

Table 4 Students' attendance at face-to-face and online classes in FY2021.

	第 1 回 (授業では第 11 回)	第 2 回 (授業では第 12 回)	第 3 回 (授業では第 13 回)	第 4 回 (授業では第 14 回)
オンライン	25	15	10	10
対面	0	8	8	12

タ端末を利用し、オンラインの受講者は、各自が所有するの PC から Zoom に接続し、原則としてリアルタイムで演習活動を行うハイフレックス型の授業形態とした。教員の PC 画面は、Zoom での画面共有を行い、同画面を PC 教室のプロジェクトにも投影した。

2020 年度および 2021 年度の毎回の授業は、Zoom の録画機能を使って収録し、演習中の不要な場面を除くなど編集したうえで PandA と連携する動画配信サービス Kaltura を用いて受講者に提供した。

なお、本科目の課題については、従来から PandA を通じて授業中の演習課題を毎回課しており、教員が採点し、コメントを付与してから、次回の授業までに受講者にフィードバックしている。

2.4.1 教員と TA の授業時のコミュニケーションについて

本科目は、2020 年度および 2021 年度の授業ではオンライン型授業を取り入れている。その際、質問対応については、オンラインの受講者は、原則として Zoom のチャット機能を利用し、対面の受講者は直接教員やティーチングアシスタント (TA) に質問するように促した。しかしながら、Zoom のチャット機能を利用する場合、教員や TA に個別に質問が送信されることがあったため、授業中の教員・TA 間のコミュニケーション手段として、本授業用の Slack チャンネルを設置し、教員と TA で質問対応の状況などを共有しながら授業を進行した。

2.4.2 演習時の質問対応について

受講者がテキスト教材を用いて各自が作業に取り組む間、受講者がつまづいた場合は、対面授業の参加者は、教室内で教員または TA (1 人) がサポートに入り、直接対応し、オンライン授業の参加者は Zoom のチャット機能を利用して対応した。また、オンライン授業の参加者の中で、画面共有を使った質問を希望した場合は、Zoom のブレイクアウトルーム機能を活用し、教員または TA が個別に対応した。

2.4.3 受講者への対応について

本科目の演習には、対面授業の場合、京都大学の教育用コンピュータ端末を用い、Python の環境には Web ブラウザ上にエディタと Python の実行環境が搭載されている「Jupyter Notebook」を用いた。2020 年度および 2021 年度は、受講者全員がノート PC を保有していたため、各自の PC に Jupyter Notebook をインストールし、起動を確認することを事前の課題とした。なお、Jupyter Notebook

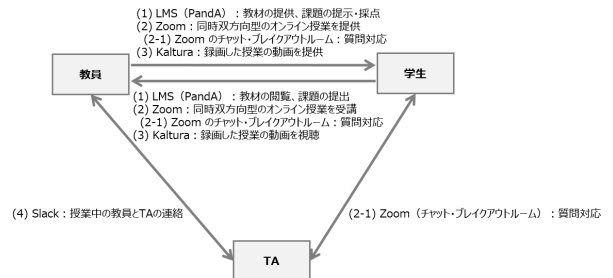


図 2 授業で活用した各ツールの役割
Fig. 2 The role of each tool used in the class.

のインストール手順書や操作手順書などは、Windows 版と macOS 版を準備した。

また、2021 年度のハイフレックス型の授業では、対面で参加する受講者は PC 教室の教育用コンピュータ端末を利用して演習を行ったが、自身のノート PC を PC 教室に持ち込んで受講する形態も認めた。

2.4.4 授業で活用した各ツールの役割について

これまで述べたように、2020 年度および 2021 年度の授業では様々なツールを利用して授業を実施した。使用した各ツールの役割について、図 2 に示す。

教員は、PandA で教材の提供、課題の提示および採点した。さらに、ビデオ会議サービス (Zoom) を利用してオンライン授業を実施し、チャットやブレイクアウトルームを活用して質問対応を実施した。加えて、Zoom で収録した動画について、不要な場面を除くなどの編集を行ったうえで、動画配信サービス Kaltura を用いて受講者に提供した。

また、教員と TA は、Slack を利用して質問対応の状況などを共有した。

一方、学生は、PandA で教材を閲覧したり、課題を提出したりした。Zoom で授業を受講し、疑問点がある場合などはチャットやブレイクアウトルームで質問した。さらに、オンデマンド型授業として、Kaltura の動画を視聴した。

2.4.5 対面授業とオンライン授業の違いについて

本実践で対象とした 2019 年度 (対面形式)、2020 年度 (完全オンライン形式) および 2021 年度 (ハイフレックス型) の授業は、各年度すべて同じ教材を使用して、可能な限り対面授業との変化をなくすように授業を実施するなど、オンライン授業でも対面授業と同等の質を担保できるように配慮した。

なお、対面授業では京都大学の演習室の PC を使用し、オンライン授業では、受講者個人のノート PC を使用した

ことから、受講生の学習環境については、対面授業とオンライン授業では差が生じた。

3. 実践結果

3.1 評価方法

本授業では、各回の授業においてプログラムの作成を課しており、その際、学習課題に応じて、順次処理、条件分岐処理 (if 文)、繰り返し処理 (for 文、while 文) に関するプログラムを作成させており、第 4 回の授業では、アルゴリズム構築からコーディングまでという一連の作業を通してプログラムを作成させている。各回で作成したこれらのプログラムについては、すべて評価の対象とした。

このほか、第 2 回から第 4 回の授業では、受講者の反応を知るため、授業の感想やコメントを自由に記述するアンケート調査を行った。

3.2 開発環境のインストールについて

本科目では、プログラミングの演習に「Jupyter Notebook」を用いた。前述したように 2019 年度の授業では、京都大学の教育用コンピュータ端末を用いたが、2020 年度と 2021 年度は個人の PC に Jupyter Notebook をインストールすることとした。インストールと Jupyter Notebook の動作確認については、事前の課題としており、表 3 に示すように本実践の対象者のうち、2020 年度は 115 名全員が、2021 年度は 26 名中 25 名が問題なくインストールすることができた。

なお、2021 年度に Jupyter Notebook がインストールができなかった 1 名については、受講者と相談したうえで Google Colaboratory を使用することにした。

3.3 受講者の学習目標について

各年度の第 1 回 (授業では第 11 回) の授業において、プログラミング学習の到達目標を尋ねた。到達目標については、「自分でプログラムを作れる程度」、「他人のプログラムを理解できる程度」、「プログラムを見て大体の構造が把握できる程度」、「プログラムの動作の仕組みが理解できる程度」、「その他」の 5 つの選択肢から選択して回答させた。その結果について図 3 に示す。

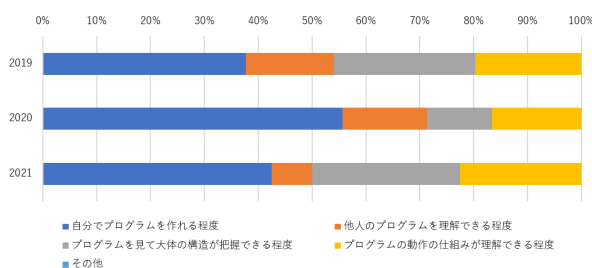


図 3 受講者の到達目標

Fig. 3 Students' own learning goals.

本授業の受講者はほとんどがプログラミング初学者であるため、「プログラムの動作の仕組みが理解できる程度」や「プログラムを見て大体の構造が把握できる程度」と回答する人がいる一方で、「自分でプログラムを作れる程度」と回答する人が約半数おり、学習目標が比較的高い受講者が多かった。

3.4 課題の提出状況について

3.1 節で述べたように、本授業では、受講者のプログラミングに関する理解度を測るため、授業ごとに課題を課している。各課題について、プログラムを完成できた (動作するプログラムを提出することができた) 受講者を「完成」、プログラムを完成できなかった (動作するプログラムを提出することができなかった) 受講者を「未完成」として集計した結果を表 5 に示す。

表中の割合は、課題提出者数に対する「未完成」の割合を示している。なお、ファイルが破損していたり、提出するファイルを間違えていたりした場合は、プログラムが完成したかどうかの判断ができないため、評価の対象としなかった。

第 1 回において、プログラムを完成できた受講者は、2019 年度は 68 名、2020 年度は 109 名、2021 年度は 20 名であった。一方、プログラムを完成できなかった受講者は、1 名、4 名、5 名であった。同様に、第 2 回において完成できた受講者は、59 名、82 名、20 名で、完成できなかった受講者は、9 名、32 名、6 名であった。第 3 回において完成できた受講者は、49 名、98 名、21 名で、完成できなかった受講者は、21 名、16 名、4 名であった。第 4 回において完成できた受講者は、43 名、94 名、22 名で、完成できなかった受講者は、14 名、14 名、3 名であった。

次に、プログラムを完成できなかった受講者の割合を見てみると、第 1 回は 2021 年度が 20.00%、第 2 回は、2020 年度が 28.07%、2021 年度が 23.08%、第 3 回は 2019 年度が 30.00%、第 4 回は 2019 年度が 24.56% と高かった。

さらに、プログラムを完成できなかった受講者のつまづき状況を把握するため、表 5 において「未完成」の受講者の具体的な誤りを確認したところ、受講者のつまづきは「文法エラー」、「論理エラー」の 2 つに大別された (表 6 参照)。なお、ここでは、文法に誤りがあり、論理にも誤りがあるものを「文法エラー」としてカウントしている。

第 1 回で文法エラーのためプログラムを完成できた受講者は、2019 年度は 1 名、2020 年度は 4 名、2021 年度は 5 名であった。第 1 回では論理エラーでつまづいた受講者はいなかった。次に、第 2 回は、文法エラーは、2 名、6 名、0 名で、論理エラーは 7 名、26 名、6 名であった。第 2 回は文法エラーよりも論理エラーでつまづく受講者が多かった。第 3 回は、文法エラーは、2 名、3 名、0 名で、論理エラーは 19 名、13 名、4 名であった。第 2 回と同様に、文法

表 5 課題の完成状況

Table 5 Number of students who completed the assignment.

	第 1 回			第 2 回			第 3 回			第 4 回		
	2019	2020	2021	2019	2020	2021	2019	2020	2021	2019	2020	2021
完成	68	109	20	59	82	20	49	98	21	43	94	22
未完成	1	4	5	9	32	6	21	16	4	14	14	3
	1.44%	3.54%	20.00%	13.24%	28.07%	23.08%	30.00%	14.04%	16.00%	24.56%	12.96%	12.00%

表 6 つまづき状況

Table 6 Situation of Mistake for each target year.

	第 1 回			第 2 回			第 3 回			第 4 回		
	2019	2020	2021	2019	2020	2021	2019	2020	2021	2019	2020	2021
文法エラー	1	4	5	2	6	0	2	3	0	0	0	0
	1.44%	3.54%	20.00%	2.94%	5.26%	0.00%	2.86%	2.63%	0.00%	0.00%	0.00%	0.00%
論理エラー	0	0	0	7	26	6	19	13	4	14	14	3
	0.00%	0.00%	0.00%	10.29%	22.81%	23.08%	27.14%	11.40%	16.00%	24.56%	12.96%	12.00%

エラーよりも論理エラーでつまづく受講者が多かった。第 4 回は、文法エラーでつまづく受講者はいなかった。論理エラーは、14 名、14 名、3 名であった。

このほか、各年度の受講者の割合を見てみると、第 1 回は 2021 年度の文法エラーが 20.00% と多い。第 2 回以降は、文法エラーより論理エラーでつまづく受講者が多く、第 2 回の 2020 年度が 22.81%、2021 年度が 23.08%、第 3 回は 2019 年度が 27.14%、第 4 回は 2019 年度が 24.56% と高かった。

3.5 受講者とその反応について

本節では、第 2 回から第 4 回までの自由記述式アンケート（授業の感想）をもとに、受講者の学習状況について示す。

3.5.1 第 2 回の受講者とその反応について

第 2 回の授業では、表 2 で示したように、2019 年度は「条件を判断して処理を実行する」、「例外処理」、2020 年度と 2021 年度はこれら 2 つに加えて「ランダムな値の生成」について学習した。

第 2 回の自由記述式アンケート結果を分類した結果を表 7 に示す。

受講者からは、「プログラミングは難しそうだという印象を持っていましたが、授業では丁寧に手順や操作方を教えていただけて、楽しいと思うようになりました」、「課題を見て、自分で『この方法を使うのかな?』と考え、試行錯誤しながらプログラミングをするのは意外に楽しいです」など、プログラミングや授業が楽しかったという記述が、2019 年度は 19 名、2020 年度は 24 名、2021 年度は 12 名からあった。さらに、「プログラミングの基礎を理解できた」、「今のところ順調に理解できています」など、授業内容の理解に関するポジティブな記述が、13 名、9 名、5 名からあった。

表 7 第 2 回の受講者とその反応について

Table 7 Students and their feedback of the second class.

	2019	2020	2021
楽しかった	19	24	12
	27.94%	21.05%	46.15%
授業の内容を理解できた	13	9	5
	19.12%	7.89%	19.23%
テキスト教材がわかりやすかった	6	8	2
	8.82%	7.02%	7.69%
次回の授業が楽しみ	5	3	0
	7.35%	2.63%	0.00%
難しかった	2	5	1
	2.94%	4.39%	3.85%

また、「テキストが非常にわかりやすくありがたいです」、「授業の内容が聞き取れなかったり授業に遅れてしまっても、プリントがわかりやすいので理解ができる」など、テキスト教材に関するポジティブ記述が、6 名、8 名、2 名からあった。加えて、「次回の授業が楽しみです!」など、次回の授業に対してポジティブな記述が、2019 年度に 5 名、2020 年度に 3 名からあった。一方、「プログラミングは初めてなので、かなり難しいと感じます」などプログラミングや課題に対するネガティブな記述が、2 名、5 名、1 名からあった。

このほか、2020 年度の受講者からは「わからないときも気軽に先生に質問できて個別に丁寧に対応してくださるのですぐにわかる」という記述が見られた一方で、「うまくいかなかった時に一人でなかなか対応できなかった」、「エラーが出たときに何行目でどういうエラーなのかを書いてはくれるが、実際に何をどう直したらいいのかを理解するのはむずかしかった」などネガティブな記述も見られた。

3.5.2 第 3 回の受講者とその反応について

第 3 回の授業では、2019 年度は「同じ処理を指定した回

数だけ繰り返す」,「条件が満たされている間に処理を繰り返す」,「ランダムな値の生成」, 2020 年度と 2021 年度は「同じ処理を指定した回数だけ繰り返す」,「条件が満たされている間に処理を繰り返す」について学習した。

第 3 回の自由記述式アンケート結果を分類した結果を表 8 に示す。

受講者からは,「とても楽しかったです」,「いろいろなプログラムを作れて楽しかったです」など, プログラミングや授業が楽しかったという記述が, 2019 年度は 8 名, 2020 年度は 10 名, 2021 年度は 4 名からあった。

さらに,「授業内容は理解できました」など授業内容の理解に関するポジティブな記述が, 16 名, 10 名, 6 名からあった。また,「課題がとても難しかった」,「前回よりもかなり難易度が上がっているように感じられた」など課題の難易度に関するネガティブな記述が, 24 名, 15 名, 7 名からあった。一方で,「課題ができたときはとても達成感があった」など, プログラムが動作した時に達成感があったというポジティブな記述が, 2019 年度に 4 名, 2020 年度に 6 名, からあった。

このほか, 2020 年度の受講者からは「私は物分かりが悪く, 授業中に課題を終わらせることがなかなかできないがプリントがわかりやすく家でもじっくり学習できるため, 最終的に理解することができます。その中でプログラミングの楽しさも理解でき, もっと学びたいと感じます」など授業外での学習についての記述が見られた。

表 8 第 3 回の受講者とその反応について

Table 8 Students and their feedback of the third class.

	2019	2020	2021
楽しかった	8 11.43%	10 8.77%	4 16.00%
授業の内容を理解できた	16 22.86%	10 8.77%	6 24.00%
課題が難しかった	24 34.29%	15 13.16%	7 28.00%
プログラムが動作した時に達成感がある	4 5.71%	6 5.26%	0 0.00%

表 9 第 4 回の受講者とその反応について

Table 9 Students and their feedback of the fourth class.

	2019	2020	2021
楽しかった	19 33.33%	26 24.07%	11 44.00%
プログラムが動作した時に達成感がある	9 15.79%	8 7.41%	3 12.00%
課題が難しかった	4 7.02%	6 5.56%	3 12.00%
プログラミングをさらに学びたい	3 5.26%	2 1.85%	1 4.00%

3.5.3 第 4 回の受講者とその反応について

第 4 回の授業では, アルゴリズム構築からコーディングまでという一連の作業を通してプログラムを作成した。

第 4 回の自由記述式アンケート結果を分類した結果を表 9 に示す。

受講者からは,「プログラミングを組み立てていくことが一種のパズルのように思えて, 楽しかった」など, プログラミングや授業が楽しかったという記述が, 2019 年度は 19 名, 2020 年度は 26 名, 2021 年度は 11 名からあった。

さらに,「完成した時はやはり達成感があった楽しかった」など,「プログラムが動作した時に達成感があった」というポジティブな記述が, 9 名, 8 名, 3 名からあった。

また,「なかなか理解することが難しかった」,「初めプログラミングをやりましたが, とても難しかったです」などプログラミング学習に関するネガティブな記述が, 4 名, 6 名, 3 名からあった。一方で,「もっと深く programming について学びたい」など, プログラミング学習に関するポジティブな記述が, 3 名, 2 名, 1 名からあった。

3.6 演習時の質問対応について

本授業では, 受講者がつまづいた場合, 対面授業では教室内で教員または TA が直接対応し, オンライン授業では, Zoom のチャット機能を利用して質問内容によって口頭あるいはチャットで回答した。また, オンライン授業では, 個別対応が必要な際は, Zoom のブレイクアウトルーム機能を活用し, 画面共有機能を使用して, 受講者の PC 画面を見ながら, 教員または TA が個別に対応した。なお, 本授業では, 受講者の質問回数に上限を設定していない。

オンライン授業を実施した 2020 年度および 2021 年度における Zoom の質問件数を表 10 に示す。2020 年度の第 1 回の授業では, ファイルが保存されている場所が分からないという質問が 22 件あった。授業時間内に教員の PC 画面を共有しながら, ファイルが保存されている場所やファイルが見つからなかった場合の検索方法について, 補足で説明を行ったが, 自分でファイルを見つけることが困難な受講者が多く, 22 件すべてをブレイクアウトルームで個別に対応した。

表 10 2020 および 2021 年度の Zoom での質問件数

Table 10 The number of questions on Zoom in FY2020 and FY2021.

		2020	2021
第 1 回	ファイルの保存場所	22	0
	Jupyter Notebook の操作方法	3	2
	エラー原因	7	4
第 2 回	エラーの原因	13	1
第 3 回	エラーの原因	31	0
第 4 回	エラーの原因	22	0

なお、2019 年度の対面授業では、Jupyter Notebook の操作方法に関する質問が散見されたが、2020 年度は 3 件、2021 年度は 2 件であった。

また、第 2 回以降は、各課題のエラーに関する質問が多く見られた。第 2 回は 13 件、第 3 回は 31 件、第 4 回は 22 件であった。このうち、第 2 回と第 3 回では、文法エラーのプログラムとエラー内容をチャットで教員に送ってきた受講者がそれぞれ 2 名ずついた。これに対して、教員は自身の開発環境でプログラムをデバッグし、エラー箇所をチャットで指摘するなど指導した。そのほかの質問に関しては、ブレイクアウトルームで受講者の PC 画面を画面共有しながら、教員または TA が個別に対応した。

なお、2021 年度はハイフレックス型の授業を実施しており、対面とオンラインの参加者数は、表 4 に示したとおりである。2020 年度の第 1 回の授業では、ファイルの保存場所が分からない受講者が多かったため、ファイルの保存場所や検索方法については、補足資料を作成し、授業時間内に説明を加えたところ、2021 年度の第 1 回の授業では、ファイルの保存場所に関する質問はなかった。第 1 回の授業は、全員がオンライン参加であったため、課題に関する質問が 4 件あったが、第 2 回は 1 件、第 3 回および第 4 回は質問がなかった。

4. 考察

本実践では、図 3 の結果から、2019 年度と 2021 年度は約 4 割、2020 年度は約半数の受講者が「自分でプログラムを作れる程度」を学習の到達目標にしており、学習目標が比較的高い受講者が多いという点において、さらに、プログラミング未経験者が受講者のほとんどを占めるという点において大きな差異が見られない集団を対象に実施されている。

4.1 受講者の開発環境について

本実践では、各年度すべて同じ教材を使用して、可能な限り対面授業との変化をなくすように授業を実施するなど、オンライン授業でも対面授業と同等の質を担保できるように配慮して授業を実施した。しかしながら、オンライン授業では、受講者個人のノート PC を使用するため、プ

ログラミングの演習に使用する「Jupyter Notebook」を個人の PC にインストールする必要が生じた。インストールと動作確認については事前の課題としていたが、2021 年度の 1 名を除き、全員が問題なく Jupyter Notebook をインストールすることができた。

プログラミングの演習をオンラインで実施する場合、受講者は自身のノート PC を使用するため、個人の PC に開発環境を構築する必要がある。PC 環境に大きなばらつきが生じる状況においては、環境構築が不要で、対面授業の演習室でも利用可能な Google Colaboratory などのクラウド環境の活用を検討することにより、教員と受講者双方の負担を軽減することが期待できる。

4.2 受講者の学習状況と反応について

本授業では、受講者のプログラミングに関する理解度を図るため、授業ごとに課題を課しており、各課題について、プログラムの完成状況を各年度で比較したところ、第 1 回においては、2021 年度が 20.00% と未完成の割合が高かった。第 2 回においては、2020 年度が 28.07%、2021 年度が 23.08% と高く、第 3 回は 2019 年度が 30.00%、第 4 回は 2019 年度が 24.56% と高かった。

ただし、対面で授業を実施した 2019 年度と完全オンラインおよびハイフレックス型で授業を実施した 2020・2021 年度では、「ランダムな値の生成」の学習回が異なるため、第 2 回と第 3 回を比較することは難しいが、オンライン授業では授業の前半でつまずく受講者が多く、対面授業では授業の後半でつまずく受講者が多く見られた。

さらに、受講生の反応について見ていくと、まず、第 2 回の自由記述式アンケートには、対面とオンラインの両方の受講者からプログラミングや授業が楽しかったという記述が、2019 年度は 19 名、2020 年度は 24 名、2021 年度は 12 名からあり、授業内容の理解に関するポジティブな記述も 13 名、9 名、5 名からあり、多くの受講生からプログラミングや授業内容についてはポジティブな評価を得られた。一方で、「プログラミングは初めてなので、かなり難しいと感じます」などプログラミングや課題に対するネガティブな記述が、2 名、5 名、1 名から見られるなど、一部の受講生にとっては、学習に不安を抱えていることが分かった。

加えて、2020 年度のオンライン授業の受講者からは、「わからないときも気軽に先生に質問できて個別に丁寧に対応してくださるのですぐにわかる」という記述が見られた一方で、同様にオンライン授業の受講者から「うまくいかなかった時に一人でなかなか対応できなかった」、「エラーが出たときに何行目でどういうエラーなのかを書いてはくれるが、実際に何をどう直したらいいのかを理解するのはむずかかった」などの記述が見られた。2021 年度の受講者からはこのような記述が見られなかったものの、今回のオンライン授業では、チャットで質問した受講者に対して

は十分な支援をすることができたが、チャットで質問しなかった（できなかった）一部の受講者にとっては、プログラムのエラーに対応することができなかつたと推察される。こうしたことから、本授業では、授業の初期段階においてオンライン受講者に対する配慮が十分ではなかつたと考えられる。これは、プログラミング初学者は、エラー表示の見方（意味）が分からないことやエラー表示の意味は理解できてもエラー箇所を見つけられないことがある [8] ため、プログラミング学習の初期段階でつまづいた際に、個人ごとに支援が求められることから、より指導を受けやすい環境の整備が必要であるという課題として浮かび上がった。

次に、第3回の授業について見ていくと、対面とオンラインの両方の受講者から「課題がとても難しかった」、「前回よりもかなり難易度が上がっているように感じられた」など、課題の難易度に関するネガティブな記述が散見されている。これは、第3回の授業で繰り返し処理を扱ったため、受講者の認知的負荷が大きくなってしまった [7] ことが考えられる。これとは対照的に、第4回では対面とオンラインの双方の受講者からポジティブな記述が多く見られており、それまでの授業内容は、学んだ内容を模倣するという比較的受動的な学習内容であったのに比べ、受講者自身でアルゴリズム構築からコーディングまでという一連の作業を通してプログラムを作成する能動的な学習部分が多いために面白く、また、同時に、達成感を感じやすかつたからではないかと思料される。いわゆる自習形態をとる能動的学習部分は、そもそも、対面式、オンラインに差が生じにくく、指導という受動的学習部分において差異が生ずるものの、学生の反応としては総体として双方が同様にポジティブな反応が多く表出したものと考えられる。

さらに、第3回の自由記述式アンケートに、オンラインの受講者から「授業中に課題を終わらせることがなかなかできないがプリントがわかりやすく家でもじっくり学習できるため、最終的に理解できます」との記述があり、授業時間外に自宅でテキスト教材を見ながら学習していることが示唆された。加えて、表10に示したように、オンライン授業では、ブレイクアウトルームで個別対応するなど、受講者が質問しやすい環境を授業の回数を重ねていく中で整えていったことから、チャットで積極的に質問していた様子が推察される。課題のプログラムの完成状況からも、オンライン授業では、授業の前半でつまづく受講者が多く見られたが、授業の後半では課題のプログラムを完成することができた割合が高く、オンライン授業では、受講者が授業時間内に積極的に質問し、個々の疑問点を解消していたと思料される。

一方、対面授業の場合は、これまで学習者の学習進捗を把握しながら授業を実施してきたつもりであったが、課題のプログラムの完成状況から、授業の後半でつまづく受講者が多く見られており、授業中に質問しなかつた（できな

かつた）受講者がいた可能性も同時に浮かび上がった。

こうしたことから、対面授業およびオンライン授業の共通の問題点として「質問のしにくさ」を、それぞれの授業形態に合わせてどのように解消していくのかについて、さらなる検討が必要であることが明らかとなった。

4.3 オンライン授業における演習時の質問対応について

本授業は、2020年度は完全オンライン型、2021年度はハイフレックス型で授業を実施した。2021年度の第1回は完全オンラインで実施しているため、他の年度と比較してみると、表10に示したように、2020年度の第1回の授業では、保存したファイルを自分で見つけることが困難な受講者が多く、すべてブレイクアウトルームで個別に対応した。

これをふまえて、ファイルの保存場所や検索方法に関する補足資料を作成し、2021年度の授業において、授業時間内に説明を加えたところ、2021年度の第1回の授業ではファイルの保存場所に関する質問はなかつた。

以上のことから考えると、オンライン授業を実施する場合、受講者は自身のノートPCを使用するため、個人ごとに設定や環境が異なることが想定される。このように、PC環境に大きなばらつきが生じる状況においては、授業時間内の教員やTAのトラブル対応の負担が大きくなる一方で、受講者もプログラミング学習以外のところに時間をとられてしまうこととなる。受講者のつまづきを軽減するためには、受講者自身で演習をして課題を提出するまでのプロセスの中で、作業を遂行するために必要となるスキルや知識に対する支援も必要であり、2020年度に用いたテキスト教材はこうした側面についての配慮が十分でなかつたと考えられる。

このほか、Jupyter Notebookの操作方法について、2019年度の対面授業では質問が散見されたが、2020年度は3件、2021年度は2件であった。オンライン授業では、受講者の様子を把握することが困難なため、開発環境の操作方法については、3回ほど繰り返して説明するように意識的に実施したことが効果的であったということが考えられる。

また、第2回以降は、各課題のエラーに関する質問が多かつた。このうち、文法エラーのプログラムとエラー内容をチャットで教員に送ってきた受講者がいたため、教員は自身の開発環境でプログラムをデバッグし、チャットで指導した。本授業では、上述の質問以外はブレイクアウトルームで個別に対応していたが、受講者の状況の把握に時間がかかってしまうことがあつた。

こうしたことから、文法エラーについては、受講者にチャットでプログラムとエラー内容を共有してもらい、教員やTAが自身の開発環境でプログラムをデバッグし、エラーを発見することで、教員と受講者双方の負担を軽減することが期待できる。

一方、論理エラーについては、文法エラーのようにエラーが明示されないことから、受講者自身でエラーを修正することが難しいため、今回実践したようにブレイクアウトルームで個別に対応することが効果的であると考えられる。

5. おわりに

本稿では、コロナ禍で実施した演習型プログラミング授業について、その実践例を報告するとともに、2019年に実施した従来型の対面式授業と比較し、演習型授業のオンライン化における課題や問題点などについて考察した。

本実践では、各年度すべて同じ教材を使用し、可能な限り対面授業との変化をなくすように授業を実施するなど、オンライン授業でも対面授業と同等の質を担保できるように配慮して授業を実施した。

オンライン授業では、受講者個人のノートPCに開発環境をインストールする必要が生じたが、ほぼ全員が問題なくインストールすることができた。オンライン授業では、PC環境に大きなばらつきが生じることが想定されるため、環境構築が不要で、対面授業の演習室でも利用可能なクラウド環境の活用を検討することにより、教員と受講者双方の負担を軽減することが期待できる。

対面で授業を実施した2019年度と完全オンラインおよびハイフレックス型で授業を実施した2020年度、2021年度の各課題について、プログラムの完成状況を各年度と比較したところ、オンライン授業では授業の前半でつまづく受講者が多く、対面授業では授業の後半でつまづく受講者が多く見られた。

今回のオンライン授業では、ブレイクアウトルームで個別対応するなど、受講者が質問しやすい環境を授業の回数を重ねていく中で整えていったことから、チャットで積極的に質問していた様子が推察される。課題のプログラムの完成状況からも、オンライン授業では、授業の前半でつまづく受講者が多く見られたが、授業の後半では課題のプログラムを完成することができた割合が高く、オンライン授業では、受講者が授業時間内に積極的に質問し、個々の疑問点を解消していたと史料される。

このように、オンライン授業では、チャットで質問した受講者に対しては十分な支援をすることができたが、チャットで質問しなかった（できなかった）一部の受講者にとっては、プログラムのエラーに対応することができなかつたと推察される。こうした点において、本授業では、オンライン受講者に対する配慮が十分ではなかつたと考えられる。一方、対面授業の場合は、これまで学習者の学習進捗を把握しながら授業を実施してきたつもりであったが、授業の後半でつまづく受講者が多く見られており、授業中に質問しなかった（できなかった）受講生がいた可能性も同時に浮かび上がった。

このほか、受講者のつまづき状況を把握するため、課題の未完成のプログラムを確認したところ、「文法エラー」と「論理エラー」の2つに大別された。論理エラーについては、文法エラーのようにエラーが明示されないことから、受講者自身でエラーを修正することが難しいため、今回実践したようにブレイクアウトルームで個別に対応することが効果的であることが示唆された。

今後は、今回得られた知見をもとに、テキスト教材を改善するとともに、対面授業およびオンライン授業の共通の問題点として「質問のしにくさ」があったのではないかと推察されたことから、対面およびオンライン授業における演習時の質問対応について検討していきたい。

謝辞 研究のためのフィールドをご提供いただいた京都大学高等教育研究開発推進センター酒井博之准教授に深く感謝申し上げます。

参考文献

- [1] 京都大学国際高等教育院：3. 全学共通科目の科目群・分野の概要（オンライン），入手先（<https://www.z.kyoto-u.ac.jp/pdf/link/link0780.pdf?1615484352>）（参照 2021-09-28）。
- [2] 岡本雅子：「主体的プログラミング学習段階」を対象としたプログラミング演習における学習困難性の考察，大学ICT推進協議会 2020 年度年次大会，pp.21-28（2020）。
- [3] 情報基礎演習：大人数に対するきめ細やかなフィードバックの工夫，京都大学高等教育研究開発推進センター CON-NECT（オンライン），入手先（https://www.highedu.kyoto-u.ac.jp/connect/topics/sakai_okamoto_kita.php）（参照 2021-09-28）。
- [4] 酒井博之，岡本雅子，日置尋久，喜多 一：一般情報教育の演習科目における受講者によるルーブリックの自己評価，大学 ICT 推進協議会 2020 年度年次大会，pp.422-427（2020）。
- [5] 喜多 一，日置尋久，中津 亨，酒井博之，岡本雅子，池田佳代，加古達也，鈴木聡介，吉川昌吾：一般情報教育における LINE-Bot 型クイズシステムの試用，大学 ICT 推進協議会 2019 年度年次大会，pp.62-69（2019）。
- [6] Columbia University: Hybrid/HyFlex Teaching & Learning, available from (<https://ctl.columbia.edu/resources-and-technology/teaching-with-technology/teaching-online/hyflex/>) (accessed 2021-09-28).
- [7] 河内谷幸子：プログラミング言語の学習法—for 文の理解に関する認知心理学的分析，言語と文化，Vol.3, pp.19-35（2006）。
- [8] 岡本雅子，村上正行，吉川直人，喜多 一：プログラミングの写経型学習過程を対象としたつまづきの分析とテキスト教材の改善—作業の自立的遂行と作業を介した理解のための支援と工夫，京都大学高等教育研究，Vol.19, pp.47-57（2013）。

付 録

A.1 テキストに掲載したサンプルプログラム

本科目で開発したテキスト教材に掲載したサンプルプログラムを図 A.1、図 A.2、図 A.3、図 A.4、図 A.5、図 A.6 に示す。

```
num = 20
num = num + 10
print(num)
num = num * 10
print(num)
num = num - 250
print(num)
num = num // 5
print(num)
num = num ** 2
print(num)
```

図 A.1 プログラムの動作する順番

Fig. A.1 The sequence in which the programs run.

```
for i in range(1, 10):
    if i % 3 == 0 and i % 5 == 0:
        print('Fizz Buzz')
    elif i % 3 == 0:
        print('Fizz')
    elif i % 5 == 0:
        print('Buzz')
    else:
        print(i)
    print('ここは for 文の中です')
print('ここは for 文の外です')
```

図 A.5 Fizz Buzz 問題 (for 文)

Fig. A.5 Fizz Buzz problem (for statement).

```
n = int(input('国語の点数を入力してください:'))
if n >= 60:
    print('合格です. ')
    print('おめでとうございます')
else:
    print('残念')
    print('不合格です')
print('処理終わり')
```

図 A.2 テストの点数の判定

Fig. A.2 Judgment of test scores.

```
c = 1
while c < 10:
    if c % 3 == 0 and c % 5 == 0:
        print('Fizz Buzz')
        break
    elif c % 3 == 0:
        print('Fizz')
    elif c % 5 == 0:
        print('Buzz')
    else:
        print(c)
    c = c + 1
```

図 A.6 Fizz Buzz 問題 (while 文)

Fig. A.6 Fizz Buzz problem (while statement).

```
n = input('好きな数字を入力してください:')
try:
    # 入力された値を int 型に変換
    n = int(n)
    if n % 12 == 0:
        # 12 の倍数のとき
        print('○')
    elif n % 4 == 0:
        # 4 の倍数のとき
        print('△')
    elif n % 2 == 0:
        # 2 の倍数のとき
        print('×')
    else:
        # どの条件にもあてはまらないとき
        print('☆')
except ValueError:
    print('数値を入力してください')
```

図 A.3 入力された数値の判定

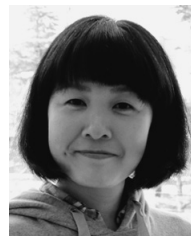
Fig. A.3 Judgment of inputted values.

```
import random

a = random.randint(0, 9)
b = int(input('好きな数値を入力してください'))
if a == b:
    print('あたり')
else:
    print('はずれ')
print('コンピュータが選択した値は', a, 'でした')
```

図 A.4 コンピュータとのじゃんけんゲーム

Fig. A.4 Rock-paper-scissors game with a computer.



岡本 雅子 (正会員)

民間企業での勤務を経て、京都大学大学院情報学研究科博士後期課程修了。博士 (情報学)。2014 年京都大学高等教育研究開発推進センター特定研究員、2014 年同特定助教を経て 2019 年より同特定講師。情報教育の研究に従事。日本教育工学会、教育システム情報学会等各会員。