

Python を用いたプログラミングスキルの基礎力習得のための 「情報機器の操作」シラバスの提案

吉田典弘^{†1} 越智徹^{†2}

概要：本報告では全ての教員免許の共通科目（66 条の 6 に定める科目）である「情報機器の操作」について、2020 年度からの学習指導要領の実施、特に小学校のプログラミング教育の必修化にも対応できる教員養成としてのシラバスを提案する。まず、この科目に関する大学の運営上における様々な問題点を示した上で、新たに一般情報教育の知識体系（GEBOK2017.1）をもとに「情報機器の操作」の 2 科目、計 2 単位のシラバスとして、アカデミック ICT リテラシーとしての科目を 1 単位、プログラミング基礎演習としての科目の 1 単位を設定する。後者に関しては、A パターンとして Python を用いて Turtle Graphics を描かせながら授業を展開するシラバス、B パターンとして Python を用いて高校数学の復習を題材としたシラバスを示す。パターン A に関しては、2019 年後期に一般情報教育レベルのプログラミングの授業において、Python を用いてプログラミングスキルの基礎力を習得することを目標とし、構造化プログラミングの基本構造（順次処理、繰り返し、条件分岐）を学習した履修学生に自由記述によるアンケートを実施した。この結果から Python が情報を専門としない学科における学生のプログラミングの学習において、分かりやすく使いやすいことと、半期 15 回の演習授業によりプログラミングスキルの基礎力が習得できる可能性があるため、提案するシラバスによる授業を実施することで、小学校のプログラミング教育を担当する教員としての素養を持たせることができることも示す。

キーワード：一般情報教育、プログラミングスキル、情報機器の操作

Operation of Information Machinery :A Syllabus to Acquire the Basic Skills of Python Programming

NORIHIRO YOSHIDA^{†1} TORU OCHI^{†2}

Abstract: “Operation of information machinery” is a syllabus that has been developed based on the curriculum guidelines to be implemented from the fiscal year 2020, and the teacher education, which can also correspond to programming educational compulsionization of an elementary school. Particularly, the knowledge of the proposed syllabus is a common prerequisite for all the teachers to obtain their licenses (the subject set as 66 articles of 6). The proposed guidelines should be applied to all the educational levels after denoting the various problems associated with these guidelines with respect to the operation of a university based on a subject (GEBOK2017.1) and another subject, which provides academic ICT literacy. This syllabus proposes two subjects of “operation of information machinery” and a total of two units, one hour per semester, and basic practice of programming. The guidelines have been implemented, and one semester hour per subject has been established as the standard. Although Turtle Graphics drawn using Python can be considered as pattern A in the latter subject, a syllabus that is developed for the class and a syllabus that contains a subject that can be used to review high-school mathematics using Python as pattern B are specified. The proposed syllabus supports the students who freely acquired basic Python programming skills via a programming session in the latter subject based on the specified target and learned the basics of structured programming (a sequential processing is repeated, i.e., conditional branch) in 2019 with respect to the pattern A based on the given description. There are a case that Python tends to use it clearly in learning of a programming of the student who can put the department which doesn't make the information a major and a possibility that practice tuition in 15 times of half term can acquire basic skills of a programming skill from this result. Finally, the proposed syllabus is needed at all the educational levels as well as for the teachers who teach programming in an elementary school.

Keywords: General Information Education, Skills of Programming, Operation of Information Machinery

1. はじめに

2020 年度から小学校のプログラミングの必修化が開始されるので、この教育を担当できる教員の育成として、現職教員へのプログラミング教育の講習などを利用する様々な方法[1]と、これから、教員となるために大学の教職課程

を履修している学生に対するプログラミング教育を実施することの 2 つが必要であると考えられる。そこで、本稿では後者に関し、全ての教員免許に共通科目（66 条の 6 に定める科目）である「情報機器の操作」について、2020 年度からの学習指導要領、特に小学校のプログラミング教育の必修化を担当できるプログラミングスキルを習得することができるシラバスを提案する。提案シラバスにおいては、情報処理学会一般情報教育委員会が作成した J17-GE の一般情報教育の知識体系（GEBOK2017.1）[2]におけるアカデ

^{†1} 羽衣国際大学
HAGOROMO University of International Studies

^{†2} 大阪工業大学
Osaka Institute of Technology

ミック ICT リテラシーエリアと、アルゴリズムとプログラミングで提唱されているユニットをベースとして、「情報機器の操作」において、大学におけるアカデミック ICT に関するスキルとプログラミング基礎演習として Python を用いたプログラミングスキルの基礎力を習得させるシラバスを提案する。Python がプログラミングスキルの基礎力を育成させるために有効な言語であるかについては、大学における一般情報教育レベルの授業において、このプログラミング言語を利用させた上で、他言語との比較や構造化プログラミングの理解に関するアンケートを実施したので、本稿で提案する「情報機器の操作」を受講することで、小学校の教員として授業を行うために必要となるプログラミングスキルの基礎力習得が可能であることも言及する。

2. 現状の「情報機器の操作」の問題点と一般情報教育の知識体系（GEBOK2017.1）の内容

本章では現在、教職課程がある大学の学部学科で実施されている「情報機器の操作」の問題点について、筆者らが過去に在籍した当時の内容と、現在在籍している大学におけるシラバスを調査したので、その結果を示す。

また、本稿で新たに提案するシラバスの前提となる、情報処理学会の J17-GE における一般情報教育の知識体系（GEBOK2017.1）のユニットについて説明し、提案するシラバスではどの部分を利用するのかを示す。

2.1 現状の教職課程必修科目「情報機器の操作」問題点

本節では、現状の「情報機器の操作」が小学校のプログラミング教育を担当できるような内容となっていないことについて、この科目を大学内で審議する組織の問題点と筆者らが所属していた、あるいは、している大学における「情報機器の操作」のシラバスを調査したので、その問題点を挙げる。理工系学部など情報通信機器を良く使う学部以外での教職課程履修者の情報活用能力は、この科目に依るはずである。残念ながら、このような非情報系学部の「情報機器の操作」は、その大学における共通教育科目で開講している場合が多く、その実情はワープロ、表計算、プレゼンに関する各ソフトウェアの操作教育が中心である場合が大変多い。そこで、この節では、筆者らが在籍した複数の大学のシラバス等を確認したので、その内容の報告を行う。

2.1.1 「情報機器の操作」のカリキュラムを検討する運営上の問題点

教職課程を実施している多くの大学では、全学共通科目の情報リテラシー教育として開講している科目、例えば「情報リテラシー」あるいは「情報処理演習」という科目の単位取得を「情報機器の操作」の単位取得として読み替えていることが多い。つまり、この科目は学部や学科専門科目

として開講されているのではないので、例えば情報系学部や情報系学科の意向、あるいは情報教育に造詣が深い教員の意向がそのまま反映されている訳ではないのが現状である。多くの大学で実施されている初年次教育としての情報教育の授業では、大学入学後の1年生前期の科目、15回分の授業として、パソコン（PC）を用いた演習形態による1単位の科目を開講し、大学のPC環境や電子メールの利用方法を教えることから始まり、情報倫理、ワープロソフトによる文書作成、表計算ソフトによる関数を利用した計算、グラフの作成、データの並べ替えや抽出について教授していることが多い。1年生後期の科目としては、15回分の授業としてパソコン(PC)を用いた演習形態による科目として、前期の科目から継続した表計算ソフトの応用としての、前期の科目で教授しなかった様々な関数の利用、クロス集計、ピボットテーブルなどが教授されている。また、大学2年次以降のゼミで必要となるプレゼンテーションソフトの利用などについても、この科目の中で学習していることが多い。大学によって差はあるが、これらの科目のカリキュラムやシラバスの作成は、情報教育の専門家が担当するのではなく、授業実施については全て開講科目の全てを外部業者に委託して、外部講師によって実施されているなど、情報教育の質の向上が十分に行えていない大学が多いのではないかと考えている。

2.1.2 教職課程のある大学における「情報機器の操作」のシラバスの調査

前項に関して実情を把握するために、筆者らが所属していた、あるいは現在所属をしている大学の「情報機器の操作」についての当時の状況と現在の状況について調査を実施した。

・S大学

文系大学であり、3学部の内、2学部に教職課程が置かれていた。筆者が在籍した当時の科目名は「情報リテラシーⅠ」と「情報リテラシーⅡ」で Microsoft Office の操作方法が中心であった。専任教員も自分の学科の学生が履修する開講科目は担当をしていたが、全学部の開講科目のほとんどを外部講師が担当をしていた。なお、2019年度のシラバスを確認したが、筆者が在籍時とシラバスの内容はほとんど変わりがなかった。

・K大学

教務機構共通教育センターの開講科目として、全学共通科目の「コンピュータ基礎」など、教養科目として複数科目を開講していた。開講科目の多くの授業内容が Microsoft Office の各ソフトウェアを使用したものが中心であった。シラバスについては全学情報教育部会という組織があり、

キャンパスにある8学部からの委員で構成されていたため、実際の授業担当者が授業内容を自由に設定できない点もあった。この部会の構成委員のほとんどが情報教育を担当していない、つまり、授業で情報系科目を担当していない教員ばかりであったのと、学部によっては教育の情報化に力を入れているところがあり、部会における発言権が大きく、その学部の意向に全学共通科目の情報教育の方針が左右されてしまうこともあった。カリキュラムの変更の検討の際には、日本学会会議による情報学の参照基準のもとにカリキュラムと授業内容を提案したが、「これは情報の専門学科のことで本学の情報教育とは関係がない」などと取り扱われることはなかった。しかし、情報処理学会の一般情報教育カリキュラムについては理解を示していた。

また、教職課程履修者が多い学部、文学部や教育学部では「情報機器の操作」に当たる科目として、「情報処理基礎」などを学部で独自に開講をしていた。これらの学部におけるシラバスを確認すると、Microsoft Officeの操作中心の教育内容であった。その理由として、全学部において2年次以降のゼミ活動で困らない程度のパソコン操作スキルが要望されていたのと、例えば教育学部の科目であっても、小中高の授業でICTを活用できるような授業とはなっていなかった。さらに、教育の情報化が進んでいる学部であっても、履修ガイドブックで確認をすると、情報処理が付いた科目の単位を取得すれば、「情報機器の操作」の単位として認定するなど、同じ大学内における学部であっても「情報機器の操作」の扱いに大きな差があった。

上記についてまとめると、K大学においては、教職課程をほとんどの学部で配置してあっても、大学全体として、小中高の教員として現場で必要とされる情報通信技術（ICT）に関する情報教育を「情報機器の操作」において実施されてはいなかったのである。なお、共通教育センターが開講する全学共通科目については、2020年度の開講科目から大幅に変更される予定である。

・H大学

「情報機器の操作」に関して、以下の表1の中から2単位以上選択必修となっている。

表1 H大学における情報機器の操作

科目区分	授業科目	単位数：選択
情報機器の 操作	コンピュータ基礎演習Ⅰ	1
	コンピュータ基礎演習Ⅱ	1
	コンピュータ応用演習Ⅰ	2
	コンピュータ応用演習Ⅱ	2

この科目の運営をしているのは、共通教育開発機構という事務組織である。入学時に履修を勧める「コンピュータ

基礎演習Ⅰ、Ⅱ」の授業内容は、1年生の前期にワープロソフトとプレゼンテーションソフトの利用、後期は表計算ソフトの利用である。担当する講師は、本学の専任教員は一人もおらず外部業者に委託をしている。また、一般情報教育の知識体系（GEBOK2017.1）に示されている教育内容やプログラミング教育については全く実施されていない。

これらの授業内容を修正あるいは変更するには事務組織に対案を示し、様々な会議で承認を得る必要があり、多大な説明と時間が必要とされる。

・0大学

工学系学部学科での「情報機器の操作」としての開講科目と授業内容について示す。0大学では、科目名が基礎情報処理で、前期の基礎情報処理ⅠではMicrosoft Officeを利用した授業を実施し、後期の基礎情報処理Ⅱでは、学科によって使用するプログラミング言語は異なっているが、プログラミングの授業を実施している。特に後期でどの学科でもプログラミングの授業を実践しているのは、これまでに述べてきた文系学部学科とは大きな違いがある。この点は本稿で提案するシラバスが目指している学習目標と同じである。

2.2 J17-GEにおける一般情報教育の知識体系（GEBOK2017.1）のユニット

本報告で提案するシラバスは、J17-GEにおける一般情報教育の知識体系（GEBOK2017.1）を元にする。この中ではカリキュラム標準の基礎となるエリア構成が策定されており、提案する「情報機器の操作」に対応するエリアとしては、GE-AILのアカデミックICTリテラシーと、GE-ALPのアルゴリズムとプログラミングがある。さらに、このエリアはユニットから構成され、ユニット毎にトピックと学習目標が定められているので、これらの全てではなく、必要と思われるものを提案シラバスでは使用する。

まず、GE-AILのアカデミックICTリテラシーは、近年の多く大学で行われている、大学における初年次教育としての情報教育がこの内容になっていると考える。このエリアでのユニットを挙げる。

GE-AIL アカデミックICTリテラシーのユニット

- ・GE-AIL1 パーソナルコンピュータ（PC）の基本的な扱い
- ・GE-AIL2 アカデミックライティングを実現する文書処理
- ・GE-AIL3 学習活動に必要なデータ処理
- ・GE-AIL4 説明技術としてのプレゼンテーション
- ・GE-AIL5 学術活動のためのインターネット利用

また、従来から大学で行われてきた操作教育と呼ばれる、タッチタイピング、ワープロソフト、表計算ソフト、プレゼンテーションソフトの利用方法、Webブラウザの利用、電子メールの使い方などは、このエリアでは先修条件：コンピュータリテラシーの項目として挙げられており、初等中等教育で扱い内容であるとしている。

次に、GE-ALPのアルゴリズムとプログラミングでは、必修としてのコア（必修）のユニットとして次の二つが挙げられている。

GE-ALPのアルゴリズムとプログラミングのコアユニット

GE-ALP1 アルゴリズム（授業時間：3コマ）

GE-ALP2 プログラミング（授業時間：4コマ）

本稿における提案シラバスでは、GE-ALP2のプログラミングに記載されている以下に挙げるトピックスと学習目標を踏襲する。

トピックス

- ・変数と制御構造等のプログラミングの構成要素
- ・プログラミング演習

学習目標

変数、制御構造等のプログラミングの構成要素を理解し、それらを含む簡単なプログラムで、学習者自らが考案したアルゴリズムを実現できる。

なお、GE-ALPのエリア説明においては、コンピュータによる具体的な問題解決方法を適切なアルゴリズムとして表現できること、このアルゴリズムに基づいて処理を行わせるように必要なプログラミング能力を修得させるとの記述があり、GE-ALP1のアルゴリズムを学習させてからGE-ALP2のプログラミングを実施することとされているが、本稿ではGE-ALP2のプログラミングを実施することで、アルゴリズムに関しても学習を行うことが出来るとしてシラバスを提案する。

3.「情報機器の操作」のシラバスの提案

この章では情報機器の操作（2単位）について、半期15回を1単位として、2科目2単位分のシラバスを示す。

3.1 「情報機器の操作」：アカデミック ICT リテラシーのシラバス

アカデミック ICTリテラシーの学習目標は、コンピュータやネットワークを大学での学術的な情報活用のためのツールとして不自由なく扱えるためのスキルを身に付けるこ

とである、特に所属する学部学科における専門科目の中でも、ゼミにおける情報収集や情報編集、プレゼンテーションなどに対応できるスキルを習得する。また、卒業論文作成や発表でも、同様なICTのスキルが必要となるので、アカデミック ICTリテラシーでは、これらに対応できる素養となる教育を実施したい。よって、この科目で扱う題材などは各大学の独自性が出て良いと思われる。

提案するシラバスを表2に示す、前提条件として GEBOK2017.1のアカデミック ICTリテラシーのユニットを利用する。半期15回の授業における最初の5回では、2.2節で示した先修条件とGE-AIL1を実施し、その後、GE-AIL2からGE-AIL5までの内容を10回分として実施する。

表2 アカデミック ICT リテラシーのシラバス

授業回数	授業内容
1～5	先修条件（高等学校までの復習） GE-AIL1 パーソナルコンピュータ（PC）の基本的な扱い
6～7	GE-AIL2 アカデミックライティングを実現する文書処理
8～9	GE-AIL3 学習活動に必要なデータ処理
10～11	GE-AIL4 説明技術としてのプレゼンテーション
11～12	GE-AIL5 学術活動のためのインターネット利用
13～14	各学生が題材を決め、GE-AIL2、GE-AIL3、GE-AIL5で学習したことを用いてレポートを作成
15	提出したレポートに関して、GE-AIL4で学んだことを用いて口頭発表

使用する教科書は、情報処理学会の一般情報教育委員会委員が演習する本[3]が出版されるとのことなので、これを待ちたい。

また、このようなアカデミック ICTリテラシーを多くの大学で行うためには、各大学で実践されている授業や、授業内での成功事例を「共有」する仕組みが必要であると考えている。1990年代の大学における一般情報教育が提唱された当時あるいは2000年前半の改訂の際には、大学の一般教育に関する授業実践報告が本研究会や多くの学会で発表されていたと思うが、近年ではGEBOKをもとにしたアカデミック ICTリテラシーの授業実践報告が少ないと思われる。

このような活動を大学の情報教育でも充実させるには、高等学校の「情報Ⅰ」に関する活動が参考となる。2019年12月から2020年1月に掛けて、神奈川県情報教育研究部会や教科書会社による情報教育セミナーなどで、先進的な取り組みをされている教員による授業実践報告や教材紹介など

を「共有」することが活発に行われており、大学における全学共通科目に関する各大学の方針で大きく違うかもしれないが、是非、授業実践の成功例を共有できる機会を設けて頂き、良い実践例を「共有」することが出来ればと考えている。

3.2 一般情報教育レベルの Python によるプログラミングの授業実践

本節では、2019年前期の授業でPythonを用いて大学の一般情報教育で使った実践報告を行う。Pythonを授業で使用するプログラミング言語として選択した理由は二つある。一つ目は2020年4月からの基本情報技術者試験の午後問題の選択言語に選定されたことである。この資格に関する過去問題集[4]には表3のように選択言語として出題されるプログラム言語の比較がある。この表から習得の容易さが窺える。

表3 選択問題で出題されるプログラム言語の比較

	C	Java	Python	アセンブラ
習得の容易さ	△	△	◎	△
実務での利用しやすさ	○	○	◎	×

二つ目としては、2025年度からの高等学校で実施される「情報Ⅰ」の教員研修資料において、プログラミング言語としてPythonが用いられていることである。

そこで、Pythonを用いたプログラミング教育が有効であるかを、2019年9月から羽衣国際大学の放送・メディア映像学科の2年生を対象とした「専門ゼミナールⅡ」（9名）でアンケート調査を実施したので結果を報告する。この授業のシラバスを表4に示す。使用した教科書は、「学生ためのPython」[5]である。この教科書を選定した理由は、プログラミングを学習する題材としてTurtle Graphicsを利用したアニメーションの作成を扱っており、筆者が所属していた大学でのプログラミングの授業で使用していたJavaの教科書ではアニメーション作成を題材としていたので、取り扱う内容が近かったことと、この授業の授業への総合的な満足度が高かったので、Pythonでもアニメーションを題材とすることは有効ではないかと考えたからである。

また、この教科書を使用中に分かったのだが、この本のまえがきには、「将来教壇に立ちプログラミングの指導を行う教育学部の学生のために作成したもの」と記述されており、このことをこの教科書を使用した授業を行うことで確認することができるのではないかと考えていた。

表4 2019年後期 専門ゼミナールⅡのシラバス

授業回数	授業内容
1	ガイダンス：Pythonでプログラムを作成し動作を確認する
2	変数と計算
3	文字列、リスト、タプル、辞書
4	制御構造を知る1 if文、while文
5	制御構造を知る2 for文、break文とContinue文
6	Turtle Graphicsを使ってみよう
7	各自でオリジナルのTurtle Graphicsを制作（1回目）
8	各自でオリジナルのTurtle Graphicsを制作（2回目）：課題を提出
9	関数を定義する、引数を使う関数
10	戻り値とは、変数のスコープ
11	関数に値を渡す方法、再帰呼び出しする関数、
12	モジュールを利用する、関数を利用した練習問題
13	関数を利用した練習問題
14	第8回で提出したTurtle Graphicsのプログラムに関する利用する（1回目）
15	第8回で提出したTurtle Graphicsのプログラムに関する利用する（2回目）：課題を提出

次にPythonが学生に取って本当に使用がしやすく、分かりやすいプログラミング言語なのかを判断するためにアンケート調査を実施した。アンケート対象学生は、情報を専門としない学科の2年生9名で、2019年後期に開講された専門教育科目「専門ゼミナールⅡ」の履修学生である、9名のうち7名は学科の情報システムコースに所属しており、1年生前期の「プログラミング演習Ⅰ」では、Visual Basicを学習している、また、1年生後期の年度末には集中講義として開講された「プログラミング演習Ⅱ」でJavaを習得している。さらに、この7名の学生は2019年の夏季休暇期間中に学外活動として、小学生にScratchを用いたプログラミング教室を開講し指導をしていた。この7名以外の残りの2名は別コースに所属している学生であり、2019年前期までにプログラミングの授業を全く受けていない学生1名とプログラミングの授業を受けたことがある学生1名であった。

アンケートは第8回のオリジナルTurtle Graphicsの課題提出後の第9回の授業の冒頭で実施した。以下に自由記述によるアンケートの質問内容を示す。

質問 1 大学入学後, Visual Basic (1 年前期), Java (1 年後期集中講義), C# (Unity を使用した授業で学習, 2 年前期と後期) でプログラミングについて学んできましたが, 9 月から Python を使用した感想を, 今までに使用したプログラミング言語と比較し, 最低 2 つ挙げてください。

質問 2 2 年の前期に Scratch を勉強しました。このようなビジュアルプログラミングと比較し, Python を使ってみてどのような点に違いがありますか? 最低 2 つ挙げてください。

質問 3 2019 年 9 月からのゼミで, 構造化プログラミングの基本構造である順次, 条件分岐, 繰り返しについて理解することが出来ましたか? 理解ができたと思う人は, 今までどの点が理解できなかったかを 2 点挙げてください。今回も理解できなかったと思う人は, 今までと同じ点が理解できなかったのか, 新たに理解できない点が出てきたのかを 2 点挙げてください。

各学生からの回答結果は付録 A に示す。9 名だけの回答であったが, このアンケート結果から分かることは以下の通りである

質問 1 への履修学生の回答から分かることは, Python は, 他の言語に比べて, 使いやすく分かりやすい傾向がある。

質問 2 への履修学生の回答から分かることは, Python は Scratch より理解はしにくい, テキストベースのプログラミング言語としては学生が受け入れやすい傾向がある。

質問 3 への履修学生の回答から分かることは, 大学入学後にプログラミングの授業を複数受講した学生であっても, 構造化プログラミングの基本構造を用いて, オリジナルのプログラムを作成する際に, これらを正確に利用できるまでの理解をしておらず, 本授業で Python を用いて復習を行ってみたが, 理解度が大幅に改善されることはなかった, よって, 構造化プログラミングの基本構造を用いて自分が意とするプログラムの作成ができるようになるには, ある程度の時間が必要であると考えられる。

先述したように, 上記のアンケートは第 8 回の授業が終了した時点のものである。表 4 のシラバスにあるように, 第 9 回以降では, 関数の利用する方法など, プログラムを短く分かりやすくする手法を学習している。最終課題は第 8 回目で提出した各自のオリジナル作品について, 第 9 回目以降で学んだ関数を用いて (引数を用いなくても構わないとした), プログラムを短くすることとした。なお, 本発表の会場では, 学生が作成した Turtle Graphics と, 最終課題においてプログラムを短く出来ていることを提示する予定である。

3.3 小学校のプログラミング教育を行う上で必要となるプログラミングのスキル

「情報機器の操作」におけるプログラミング基礎演習としてのシラバスの提案を行うが, 到達目標を設定する上で, 文部科学省が示している小学校段階のプログラミングに関する学習活動分類を参考にすることとした。その内容は以下の通りである。

- 小学校段階のプログラミングに関する学習活動の分類
- A 学習指導要領に例示されている単元等で実施するもの
 - B 学習指導要領に例示されていないが, 学習指導要領に示される各教科
 - C 教育課程内で各教科等とは別に実施するもの
 - D クラブ活動など, 特定の児童を対象として, 教育課程内で実施するもの
 - E 学校を会場とするが, 教育課程外のもの
 - F 学校外でのプログラミングの学習機会等の内容を指導する中で実施するもの

上記の活動のどこまでを担当できるか, あるいは担当すべきかを検討したが, 小学校免許の教職課程がある学部学科における「情報機器の操作」では, 学習活動の A 分類と B 分類までを担当できるような内容であることが望ましいと考える。B 分類は学習指導要領に示されていないが, 学習指導要領に示される各教科へのプログラミングを導入するので, A 分類は確実に出来るようにしおき, そのノウハウを B 分類に応用できるような素養を身に付けることができればと考えている。F 分類までを担当できるように大学の情報教育で実施をしたいが, 学科の専門教育科目において, 「情報機器の操作」で学んだことを基礎とした継続の科目を開講して F 分類まで対応できる情報教育を実施することが望まれる。

3.4 「情報機器の操作」: プログラミング基礎演習 (A パターン) のシラバス

前節までのことを踏まえ, 「情報機器の操作」の後半の科目として, プログラミング基礎演習を実施するためのシラバスを提案する。表 5 は Python を用いた半期 15 回 1 単位のシラバスである。これは 3.2 節で示した表 4 の授業内容と同じであり, Turtle Graphics を題材として, 構造化プログラミングの基本構造を理解することと, 配列や関数を用いて, 短く分かりやすいプログラムの作成方法を習得することを目的としている。この授業で使用する教科書は, 学生のための Python[5] である。なお, このシラバスをプログラミング基礎演習の A パターンとする。

表5 プログラミング基礎演習（Aパターン）のシラバス

授業回数	授業内容
1	ガイダンス：Pythonでプログラムを作成し動作を確認する
2	変数と計算
3	文字列、リスト、タプル、辞書
4	制御構造を知る1 if文、while文
5	制御構造を知る2 for文、break文とContinue文
6	Turtle Graphicsを使ってみよう
7	各自でオリジナルのTurtle Graphicsを制作（1回目）
8	各自でオリジナルのTurtle Graphicsを制作（2回目）：課題を提出
9	関数を定義する、引数を使う関数
10	戻り値とは、変数のスコープ
11	関数に値を渡す方法、再帰呼び出しする関数、
12	モジュールを利用する、関数を利用した練習問題
13	関数を利用した練習問題
14	第8回で提出したTurtle Graphicsのプログラムに関する利用する（1回目）
15	第8回で提出したTurtle Graphicsのプログラムに関する利用する（2回目）：課題を提出

3.5 「情報機器の操作」：プログラミング基礎演習（Bパターン）のシラバス

ここでは前節の教育内容とは異なるプログラミング基礎演習（Bパターン）を提案する、この科目もAパターンと同じPythonを用いた半期15回で実施する演習科目であり、シラバスを表6に示す。この科目の特徴は、高等学校の数学の復習をPythonによって行う点である。前述のAパターンと違うところは、学生が高等学校で学んだ数学という科目の復習を行うことと、ある科目を学ぶ時にプログラミングを用いた手法による学習方法を体験させることである。また、扱う内容に統計も含まれるので、プログラミングを利用した統計の理解についても触れることができる。この授業での体験が現場の教員として、各教科の授業を担当した時に、何らかのヒントになることを期待したい。

なお、2020年4月からの羽衣国際大学の放送・メディア映像学科での専門教育科目の授業で、このシラバスによる授業を実施する予定である。使用する教科書は、文系プログラマーのためのPythonで学び直す高校数学[6]である。

表6 プログラミング基礎演習（Bパターン）のシラバス

授業回数	授業内容
1	ガイダンス：Pythonでプログラムを作成し動作を確認する
2	コンピュータの演算 四則演算、シフト演算
3	ビット演算、論理演算
4	方程式で図形を描く matplotlibでグラフを描く、方程式からグラフへ
5	直線の方程式、比例式と三角比
6	三平方の定理、便利な公式
7	行列とは、行列の演算
8	図形の一次変換
9	集合、順列と組み合わせ
10	確率
11	統計とは、ばらつきを調べる（分散と標準偏差、偏差値）
12	関係を調べる（散布図、共分散と相関係数）
13	データから推測する（移動平均、回帰直線）、ランダムに値を選ぶ
14～15	最終課題：小学校高学年の算数でPythonの利用を想定した授業の提案レポート

4.考察

2022年度から高等学校で「情報Ⅰ」が実施されるので、第3章で提案したシラバスは2024年度までの大学入学者を対象とした一般情報教育に適用できると考える、そこで本章では、2024年度まで大学における一般情報教育と2025年度以降の大学における一般情報教育の在り方について、プログラミング教育を大学の初年次教育として実施するという観点から考察する。

4.1 2024年度までの大学における一般情報教育

本報告が公開される時点で、2020年4月から各大学における情報教育に関するシラバスを変更することは出来ないが、2021年度からの「情報機器の操作」として読み替えている科目については、以下のような内容で開講されることを期待したい。まず、3.3節で提案したように、大学入学後の1単位科目として、アカデミックICTリテラシーによる教育を実践して頂きたい。ここでは、先修条件について、どこまで授業時間を充てるかが問題となってくるはずである。しかし、高等学校の「情報Ⅰ」における教育によって、大学入学者の情報活用能力が向上することを期待するので、必要とされる授業時間は少なくなると思われる。

次に、3.4節と3.5節で示したようにアカデミックICTリテラシーの実施後のシラバスとして、プログラミング基礎演習の2つのパターンを提案したが、この授業の題材とし

では、プログラミング教育では従来からアニメーションなどを扱うことが学習者の理解を深めることに有効であるとされていたが、本報告のアンケート結果でも示しているように、PythonによるTurtle Graphicsを題材とした場合でも同様にプログラミングの授業が楽しく思えることと、構造化プログラミングを基本構造については理解が深まるような可能性があると考えている。

授業アンケートの対象が9名という少数であることと、アンケートの対象とした学生が、2年生前期終了までにいくつかのプログラミング教育の授業を既に履修していたが、授業担当者の実感としては、2019年後期の授業開始時には、構造化プログラミングとは何かについても理解をしていない様子であったので、本調査が初学者にプログラミングを教授する場合に、Pythonを利用した場合に学習者がどのように考えるか、その傾向を示せたはずである。

つまり、Turtle Graphicsなどの題材を利用すれば、構造化プログラミングの基本構造、配列、関数の利用について、大学の一般情報教育の授業を受講するまでに、プログラミング教育を全く受けていない場合であっても、半期の授業を受講することで、自分でオリジナルのプログラミングを作成ができるようになるまで指導ができ、さらに関数や配列などを学習させることで、オリジナルのプログラムを短く分かりやすいものに修正をさせる体験もできるので、プログラミングスキル基礎力の一部分ではあるが体得させることができるものと考えている。

4.2 2025年度以降の大学の一般情報教育

本報告で提案したシラバスの制約事項という観点から論じる。2022年度から実施される高等学校での学習指導要領における「情報Ⅰ」の実施により、プログラミング教育が高校生全員に指導される。これらの教育を受けた生徒が大学に入学してくる2025年度からの大学における一般情報教育では何を教授すべきかを検討したい。高等学校までのプログラミング教育をきちんと習得していない、簡単に言えば構造化プログラミングの基本構造を理解していない入学生に対しては、復習のための授業を行っていく必要が出てくることが予想されるが、高等学校で構造化プログラミングの基本構造について、少しでも触れていれば大学の情報教育で復習に割く時間は少ないと考えている。このことは学生へのアンケートの質問1の回答からも予測できる。

高等学校でのプログラミング教育がより充実した場合には、本稿で提案した「プログラミング基礎演習Ⅰ」であれば、この発展科目として「プログラミング基礎演習Ⅱ」と位置付けられるものを大学の授業に用意すべきである。この科目では、教育用言語としてPythonを選択するのであれば、様々なライブラリを利用してゲーム制作やスマートフォンサイトの制作などを行うことが可能である。あるいはアルゴリズムの理解を深めるために線形探索、二分探索

などから始め、ツリー構造、ハッシュ法、整列など基本情報技術者試験の午後試験で必要となる項目を取り上げてても良いと考える。この時には、GE-ALP1のアルゴリズムのユニットをもとに授業を展開すべきであろう。

上記の科目は、提案するプログラミングの授業における発展科目としての位置付けとなるものであるが、一方、これからの技術革新に対応した「プログラミング基礎演習Ⅱ」も検討しておくべきである。人工知能(AI)に関するプログラミングを一般情報教育レベルでも教授することがひつようとなってくるはずである。既にGEBOK2017.1では、GE-AIDとして人工知能(AI)とデータ科学のエリアが提唱されている。AIのプログラミングでは、機械学習(教師あり学習)やディープラーニングなど、プログラミングをする前段階で大量のデータを事前に読み込ませるなど、構造化プログラミングとは異なる仕組みの理解が必要となってくるはずである。その先には2019年10月に量子超越として話題となった量子コンピュータについても、その原理を一般情報教育の中で理解させることが必要となってくるはずである。AIや量子コンピュータについては、技術革新によって、これらを学習するためのサポートシステム等が開発されていくはずだが、これらの技術の根底を支えるのは数学や量子力学(この学問を理解するにも数学のスキルが必要であると言われている)である。早くても2035年頃になるかもしれないが、この時の一般情報教育では大学の担当教員が数学を苦手とする学生にも対応し、新技術を分かりやすく、ワクワクした気持ちで楽しい授業を展開できる準備をしておく必要があると考えている。

5.おわりに

本報告では、現状の教職課程の必修科目である「情報機器の操作」の様々な問題点を示した上で、新たに一般情報教育の概念であるGEBOK2017.1を元に「情報機器の操作」の2科目、2単位のシラバスとして、アカデミックICTリテラシーの科目を1単位、プログラミング基礎演習の科目の1単位に関しては、AパターンとしてPythonを用いたTurtle Graphicsを描かせるシラバスと、BパターンとしてPythonを用いた高校数学のやり直しを題材としたシラバスの提案をした。Aパターンに関しては、2019年後期の大学におけるプログラミングの授業において、履修学生にアンケートを実施したことで、情報を専門としない学科の学生に取ってPythonが分かりやすく使いやすいこと、また、プログラミングスキル基礎力の習得として、構造化プログラミングの基本構造(順次処理、繰り返し、条件分岐)の理解が半期15回の授業実施でも深まる可能性があることを示した。さらに、自分が書いたプログラムに関数を利用させることで、プログラムが短く分かりやすいものに出来ることを体験させることが出来た。今後の課題は、Aパターンに関しては、

2020年4月からの1年生の授業「プログラミング演習Ⅰ」において、同様のシラバスによって授業を実践するので、定量的な結果が示せるように、事前事後アンケート調査を実施することと、Bパターンについても2020年年4月からの別の授業において実践を行うので、Aパターンと同様の事前事後アンケート調査を実施する予定である、これらの結果から提案シラバスによる授業の有用性および両パターンの結果の比較を行い、どちらのシラバスが大学初年次のプログラミング基礎演習としてふさわしい内容であるかも検討したい。これらの活動により、大学の教職課程履修者の情報活用能力が向上するとともに、大学における一般情報教育の教育内容の改善にも利用されることを期待する。

謝辞 Pythonの利用に関するアンケート調査に回答をしてくれ、た、羽衣国際大学現代社会学部放送・メディア映像学科、情報システムコース2年のゼミ生に感謝する。

参考文献

- [1] “みんなのコード【教員向け】教員養成への取り組み” .
<https://code.or.jp/school/eb120dec563>, (参照 2020-1-22).
- [2] “カリキュラム標準一般情報処理教育 (GE)” .
https://www.ipsj.or.jp/annai/committee/education/j07/ed_j17-GE.html, (参照 2020-1-22).
- [3] 木野富士男著, 高橋尚子監修.[演習]アカデミックスキルとしての ICT 活用.技術評論社,2020.3 発行予定.
- [4] 株式会社ノマドワークス. かんたん合格 基本情報技術者過去問題集令和 2 年度春季.インプレス,2019,9p.
- [5] 本郷健, 松田晃一. 学生のための Python. 東京電機大学出版局,2017.
- [6] 谷尻かおり. 文系プログラマーのための Python で学び直す高校数学.日経 BP 社,2019

付録

付録 A Python による授業を受講した学生の自由記述によるアンケートの質問と回答

質問 1 大学入学後, Visual C (1 年後期), Java (1 年集中講義), C# (Unity で使用 2 年前期と後期) でプログラミングについて学んできましたが, 9 月から Python を使用した感想を, 今までに使用したプログラミング言語と比較して書いてください. 最低 2 個, 挙げてください.

回答

- 学生 A 短くて分かりやすい. 保存と実行の方法が他のプログラムと大きく違うと感じました.
- 学生 B 難しくて, ややこしいです. 特に実行するときです. しかし, フォントは見やすいです.
- 学生 C 圧倒的にプログラムを書く量が少なくて済む.
Java が苦手でも理解しやすい
- 学生 D 使いやすい, 短い, 楽.
- 学生 E 絵を描くのが楽しかった. 初めてプログラミングを学んだが, 印象は難しいと思います.
- 学生 F 他のプログラムよりシンプルで使いやすいと思います. また, 自分が今どんなプログラムを作って何をしているのか分かりやすいと思いました. これは私が悪いですが, 間違っているところを見つけるのが少し難しく感じました.
- 学生 G Java よりもプログラムが短くて不安になるくらいでした. 自分にとって Python では追加で覚えることが少ないので好きだと思いました.
- 学生 H 簡単にできるのは楽でよかった. プログラミングの仕組みを理解するのはどれを使っても難しいと感じた.
- 学生 I Java が得意じゃなくても Python は分かりやすくできる. 打つ構文が圧倒的に少ない.

質問 2 2 年の前期に Scratch を勉強しました. このようなビジュアルプログラミングと比較し, Python を使ってみてどのような点に違いがありますか? 最低 2 個, 挙げてください.

回答

- 学生 A Scratch では基礎構造のみを使ったプログラムだとどうしても長くなってしまふ. オブジェクトごとにプログラムを設定するため全体の構造がわかり難くなる.
- 学生 B まずは言語の違い点です. 実行する時に Scratch の方はわかりやすく, コードのどこかが間違ったかが分かります. Python はその点が分かりづら

いと思います。

学生 C あらかじめ用意されている物を組む Scratch に対して Python はタートルグラフィックスを用いている。Scratch はパネルを組むが Python は Java で学んだことを使う。

学生 D Python は文とパネルによる表示であるため、Scratch の方が構造は分かりやすい。Scratch はなんとなくの感じで作れる。

学生 E Scratch をやっていなかったのでは判断は出来ません。

学生 F Python ではプログラムを書いて作っていきますが、Scratch ではもともと用意されているプログラムのブロックなどを組み合わせて作るので Scratch の方が少し作成の工程が短縮されています。

Scratch はゲーム作りが簡単にできるので、プログラムの結果を楽しく理解できます。しかし、プログラミングを自分で書いていくわけではないので他のプログラミング言語へ移行する場合には Python と比べると難しいと思います。

学生 G Python は自分で命令文を入力していく。また、どこにエラーが出ているか Python では分かりやすい。

学生 H Scratch は使ったことはありません

学生 I Scratch はある程度決められた構文で作る。Python に比べてプログラムが長くなる。Scratch は独自のパネルを使うが Python は構文を使う。

ん。作った変数に処理を行った後、数字を足す方法で繰り返す回数を指定したかったのですが、足し方がきちんと理解できません。それと、if 文と while 文を組み合わせて自分のプログラムを作成する方法がわかりません。

学生 G 特に変わらなかったです。

学生 H 教科書を見ないと何しているかわからない。自分で1から作るまでには理解できていない。

学生 I まだ、完璧に理解できていないです。特に関数の利用が曖昧です。繰り返しや if 文も完璧ではありませんが Python になって少し理解ができた。

質問 3 2019 年 9 月からのゼミで、構造化プログラミングの基本構造である順次、条件分岐、繰り返しについて、理解することが出来ましたか？理解ができたと思う人は、今まではどの点が理解できなかったか 2 点、挙げてください。今回も理解できなかったと思う人は、今までと同じ点が理解できなかったのか、新たに理解できない点が出てきたのか、2 点挙げてください。

回答

学生 A 今回理解できていない点は、リスト、タプル、辞書の使い方が覚え切れていないことです。

学生 B 授業の進み方が少し早いかなと思っています。だから、理解できない点はもちろんあります。文字コードの記号とか、まだ覚えていません。

学生 C 繰り返しである for 文の仕組みについて、今一理解が出来ていなかったのですが、Python で復習してから前より理解することができた。if 文については少しまだ理解できない。

学生 D 今までも理解できていた。

学生 E 教科書を見ないと課題ができないです

学生 F 繰り返しの文を書くのが未だに理解できていませ