

電気通信大学における 「コンピュータリテラシー」科目

久野 靖

電気通信大学

大学における1年次情報教育

大学のカリキュラムの中で、1年次情報教育の位置づけと内容は大学によってかなりバラバラである。1年次情報教育とは、専門に分かれる前の情報教育ということで、「一般情報(処理)教育」とも呼ばれる。本会も情報処理教育委員会の下に一般情報教育委員会を持ち、一般情報処理教育の知識体系(GEBOK)や一般情報処理教育カリキュラムの策定を行っているが、これらが一般に広く普及しているという状況では(残念ながら)ない。

そして筆者が見聞してきた範囲では、各大学の1年次情報教育は、その設計担当になった者(大体は情報系の教員)がいろいろ苦心して作り上げている。そのため、その大学の方針や事情と担当者のアイデアや主義主張が色濃く反映され、それゆえにバラバラなのだと思う。特に後者については、上述の通り担当が大変苦勞する仕事なので、自分の主義くらい盛り込まなければやってられません、ということなのだろう。

また、その「バラバラ」を調査するという大事業が文部科学省からの委託により2016年度に本会により行われており(一般情報教育委員会のメンバが多く作業されている)、その結果のあらましが文献1)などで読める(調査そのものは専門教育も含んだ情報学全般が範囲となっている)。

この解説には各大学がどの項目に多くのエフォートを割り当てているかが掲載されているが、それによると「コンピュータリテラシー」「情報倫理とセ

キュリティ」「情報ネットワーク」がトップ3であり、その後は「情報のデジタル化」「コンピューティング」「情報システム」「コミュニケーション」が横並びで続き、「アルゴリズムとプログラミング」「データモデリングと操作」が下位、となっている。なんだけ、文系っぽい「社会と情報」が「情報の科学」を抑えて多数を占める高校情報科の現状と類似して見える気がするが、どうだろうか。

と、人ごとのように書いてきたが、本稿もその「バラバラの1つ」の紹介である。なぜということになるが、電気通信大学は情報系がかなり大きな比率を占める理工系の単科大学であり、そのこともあって上で述べた一般的傾向(文系っぽい初年次情報教育)とはまた違うカラーの内容が実現できていると思うからである。

名称はコンピュータリテラシーだけど……

本稿で紹介する科目は、1年前期に2単位の必修科目として開講される^{☆1}その名もずばり「コンピュータリテラシー」だが、その内容は他大学の同名科目とはかなり違う。そのシラバスから特徴を紹介する。まず科目の目標は次のように定めている。

達成目標：コンピュータの基本的な構成と Unix という OS の基本を学び、情報倫理、情報セキュリティについて理解することと、実際にコンピュータを道具として使いこなせるようになること

^{☆1} 後期には同じく2単位必修で「基礎プログラミングおよび演習」がある。

「Unix という OS」というところで「ん？」となるが、それ以外は一見普通そうである。しかしさらに進んで、単位を取得する上での最低基準になると表-1の通りである。

これを読んでどのように思われるだろうか。筆者がはじめて見たときは^{☆2}、なんだか、言ったら申し訳ないけれど、1980年代の情報系の入門科目のような懐しさを覚えた。もちろん当時WWWやHTMLはなかったし、著作権や情報社会が入って今風になっているが、それにしてもである。

筆者はUnix大好きで、情報系なら当然Unixをマスターして使いこなすべしと思っていたが、情報系以外の専攻もある大学で、1年生が会える最初の情報科目（そして必修）がこの内容というのは少しびっくりした。一方で、これがきちんとできるなら大変すばらしく、ぜひそれをお手伝いしたいと思ったことも記憶している。

実装の難しさ

科目の内容には賛成だが、問題はその学習をどのように成立させるかである。昔の情報専門学科であれば、PCはまだおもちゃであり、研究をするのも論文を書くのも学科の共用システムでやるしかなく、学生

表-1 コンピュータリテラシーの単位取得基準

情報基盤センター利用条件を理解し、計算機および教材へのログイン、ログアウトができる。
コンピュータネットワークやWebの仕組みの概要を理解している。
情報化社会において被害者、加害者とならないための知識を持っている。
電子メールの読み書きができ、マナーを身につけている。
Unixの基本コマンドや、ファイルとディレクトリの基本概念を理解している。
エディタで文書の編集ができる。
計算機の基本構成やその構成要素の機能を理解し、さらに値の表現方法の基礎を理解している。
OSの基本構成やプロセスなどの基本概念を理解している。
WWWで情報の検索を行うことができる。
著作権や剽窃について理解をしている。
htmlの概要を理解し、簡単なWebページの作成ができる。
文書整形システムLaTeXでの処理の概要を理解し、簡単な文章の清書ができる。

^{☆2} このシラバスは筆者が2016年度に1年次情報教育の担当として着任する前から引き継がれてきたもので、2016年度から改組だったため、当分は変えられないという状況である。

は当たり前のようにCPUの原理やOS、コマンド、エディタなどを学んでいたが、今は時代が違う。「便利なGUI^{☆3}があるのになんでこんな不便なコマンドとか打たされるの」と思われたら負けである。

前任者が構築した科目運用では、この問題に対する解答は「これらのことを大学生が学ぶべき知識として扱う」ことであった。確かに大学ではさまざまな知識を学ぶので、その一環という見え方は分かりやすいし学生にも納得が得られやすい。教員がファイルシステムやプロセスやUnixコマンドやLaTeX、HTMLの書き方について講義し、学生はLMS上の小テストで知識を確認し、試験で正解して単位を取る。

しかしこの方法では、学生は「普段自分がいろいろなことをするのにこの科目で学んだことを使おう」とは思わないだろう。それではリテラシー（読み書き能力）とはいえない。そうではなく、学生を普通のGUIとWYSIWYG^{☆4}から引き離してCUIやマークアップを体験させ、最後には「こちらのやり方も良い点がある」と納得してもらいたい。

これらの課題に対応できるような科目の実装をどのようにするかが、設計時の重要目標となった。

科目の実装方針

最終的に筆者らが選択した本科目の実装方針は次の通りである。

- 授業時間中に演習をメインにする — 演習科目が別にあるわけではないので、この科目の時間中に学生にできるだけシステムを操作してもらおうようにする。
- 予習を前提にする — 授業時間中に講義をしていては演習の時間が残らないので、テキストとそれを説明する講義ビデオを用意し、予習を義務づける。
- レポートを毎週課す — 授業時間中だけでは演

^{☆3} Graphical User Interface. 対義語がCUI, Character User Interface.

^{☆4} What You See Is What You Get. 画面で最終出力と同じものが常に見えた状態で編集する方式。対義語がマークアップ（入力で整形などの指示を混ぜて指定する）。



習の量が限られるので、テキスト中に演習課題を用意し、そこから1つ以上選択して課外に実施し、レポート (assignment) として提出させる。さらにその準備として毎回授業終了時に当日の報告 (activity-report) を提出させる。これらにより毎週文章を書かせることで書く練習ともなる。

- 多様な課題の提供 — 課題は毎回9程度用意し (典型的には演習1, 2, 3がそれぞれ a, b, c の小問を持つ形), 難易度に大きく幅を持たせた上で「1個以上を選んで提出」とすることで学生のレベルや関心の多様性に対応する。
- LaTeXによるレポート — LaTeXを扱う#9から最終回手前の#14まではLaTeXでのレポート作成を義務づけ、LaTeXを実用にする機会を強制的に設ける。
- 担当教員によるレポート評価 — assignmentは担当教員が評価し、ABC (それぞれ4, 3, 2点) で評価をつけ^{☆5}, なるべくコメントも返してもらいフィードバックの機会を設ける。
- レポートの評価はBを通常点とする — すべてBで51点となり、試験点と半々で成績をつけることで演習・課題重視を明示する。またAの比率はごく少なくし、基本的にBとすることで評価の負担を減らす。
- 試験は文章の正否を問う問題に加えて短冊問題^{☆6}を出題 — 短冊問題はもともとプログラミングの問題を穴埋め (パターンで正解されやすい) や自由記述 (採点が大変) 以外の形で出題するために開発したものだが、ここではプログラムに加えて「コマンドパイプラインの並び」「PPM (Portable PixMap) 画像のASCII表現」「HTMLソース」「LaTeXソース」などにも適用し、知識のみでない実践の必要性を訴求する。

☆5 15回中2回は総合課題回とし、配点を倍にしている。

☆6 筆者らが情報入試研究会で情報科の試験問題作成用に開発してきた出題形式で、選択肢を画面上で並べ替えて正解を構成させる。

各回の内容

表-2に各回の内容を示す。#1は初回なのでガイダンスと新入生テストの時間が多く、あとはUnixにログインするのと、安全なパスワードの作りかた、パスワード変更が内容となっている。#2はUnixに慣れることに主眼があるが、pingなどのコマンドを動かし、あとグループになって紙に文字を書いてパケットとして送りあい、エラー制御のしかたなどを自分で考えてもらう (グループワークの練習という意味もある)。#3はセキュリティや暗号の話題もあるが、実習としてはブラウザで証明書やPKIの連鎖を見ることと、あとThunderbirdを設定してメールの送受を行い、ヘッダをしてみる実習がある。#4はコンピュータの原理として、簡単なCPUのシミュレータをブラウザ上で動作させ、アセンブリ言語でプログラムを書く実習をしてもらう²⁾。

#5から本格的にUnixの内容となり、ファイル階層やパス名と保護設定の実習を行う。#6ではEmacsを用いてテキストエディタを扱い、またファイルの文字コードをダンプで調べるなどの実習がある。#7はOSの役割で、Unixのプロセス観察や自分のプロセスの停止/再開などの実習を行う。#8は代表的なフィルタや正規表現を扱う実習がメインとなっている。

#9からLaTeXに入り、以後#14までレポートはLaTeXで書いてもらうようになっている。#9はLaTeXの基本で、#10は画像 (ピクセル画像、ベ

表-2 各回の内容

#1 コンピュータの利用と認証
#2 インターネットの原理
#3 ネットワークと安全性
#4 コンピュータの動作原理
#5 ファイルシステムとファイル操作
#6 テキストファイルとエディタ
#7 コンピュータシステムとOS
#8 フィルタとシェルスクリプト
#9 マークアップによるテキスト整形
#10 グラフィクス/図と表
#11 アカデミックリテラシ (総合実習)
#12 HTML / CSSによるWebページ記述
#13 Webと情報アーキテクチャ
#14 Webサイトの設計/製作 (総合実習)
#15 ソフトウェア開発とテストケース

クター画像)について実習したあと、図(と表)を LaTeX に含める実習まで行う。#11 はアカデミックリテラシと称しているが内容は総合実習で、学生に3~5名のグループになってもらい、好きに決めたテーマでディスカッションをしてもらう。その結果を各自が LaTeX でレポートとしてまとめて提出するので、他人の発言なども報告するきちんとしたレポートを書くことに主眼がある。

#12 から Web 制作に入り、まず HTML と CSS によるページ記述を学ぶ。この回の実習はブラウザ上で HTML を打ち込むとすぐプレビューが見られる実習ページ(図-1)で主に実施し、最後にそれをファイルに保存して学内専用のサーバで見る。#13 は複数ページをリンクしたり画像を入れるのでファイルに HTML+CSS を書く形での実習になる。これらが済んだあと、#14 で再び3~5人のグループを組み、総合実習として Web サイトの設計/制作を行い、結果をレポートとして報告してもらう。

最後は落穂拾い的内容(実質は#4の続き)で、JavaScriptにより高水準言語のプログラムを体験し、またソフトウェア開発、テストケースなどの考えも学ぶ。この回も JavaScript を実行したりテストケースを実行する実習ページを主に使って演習する。

2018年度のように

筆者は直接の担当は1クラスだけであるが、責任者としてできるだけ多く授業の見学に出向き、ようすを見た。#1~#3はやさしい内容であり学生も楽しそうにやっていた。#4のアセンブリ言語プログラミングは、プログラミング初心者から多く「プログラミング体験ができよかった」という感想が聞かれた²⁾。

このように出だしはよかったが、#5になるとGUIを使わずコマンドでファイルを操作することに大きなとまどいと反発が見られた。その後、#6、#7、#8と毎回 Unix(や Emacs)の新しい内容が出て来るので大変そうでもあった。

続いて#9で LaTeX に入るとマークアップに対するとまどいや Word の WYSIWIG の方がよいという反発が多く見られた。#10は PPM や PostScript をエディタで打ち込むのがメインなのでそこは楽しそうにやっていたが、図を LaTeX にうまく埋め込めない学生が見られた。以後#14まで毎回 LaTeX によるレポートなので、画像が埋め込めない人数は減っていつている(できても面倒なので入れないという学生はいた)。#11は総合課題としてディスカッションを求めたが、そのようなことは初めてという学生も多かった。しかし最終的には「討論は新しいアイデアが出る良い方法だ」というコメントが多く見られた。

#12で HTML+CSS に入ると、演習ページによるプレビューが簡単なのと、Web ページそのものに親しみがあることから、好意的な感想が多くなった。#13ではファイルで HTML を扱う必要があったが、ほかのページへのリンクや画像の参照(埋め込み画像、背景画像)ができることのインパクトが強く、引続き好意的な感想が多かった。#14はグループで Web サイトを制作するので、グループによって協力がうまくいったところは良い体験だがそうでないグループもあった。しかし全体的には面白かったという感想が多かった。最後の#15は#4との間が空いているので心配されたが、思ったより楽しそうに JavaScript で実習をしていた印象である。

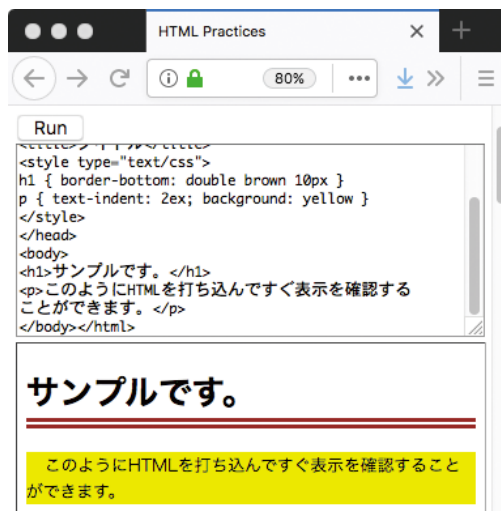


図-1 HTML 練習ページの画面



表-3 どれくらい新たなことを学んだか？

多数学んだ	まあ学んだ	どちらでもない	少しだけ	まったくくない	未回答
405	243	33	18	5	10
56.7%	34.0%	4.6%	2.5%	0.7%	1.4%

n=714

表-4 課外に平均してどれくらい時間を使ったか？

30分以下	1時間以下	2時間以下	4時間以下	4時間超	未回答
49	104	288	184	79	10
6.9%	14.6%	40.3%	25.8%	11.1%	1.4%

n=714

表-5 この科目の負担はどれくらいに感じたか？

過小	楽勝	適切	やや大	過大	未回答
8	5	245	324	122	10
1.1%	0.7%	34.3%	45.4%	17.1%	1.4%

n=714

表-6 この科目の内容の難易度はどうか？

やさしすぎ	やややさしい	適切	やや難しい	難しすぎ	未回答
6	14	209	353	122	10
0.8%	2.0%	29.3%	49.4%	17.1%	1.4%

n=714

表-7 内容の全体的な分量はどうか？

少ない	やや少ない	適切	やや多い	多すぎ	未回答
4	12	235	338	114	11
0.6%	1.7%	32.9%	47.3%	16.0%	1.5%

n=714

表-8 この科目が必修でなかったとしたら取るか？

絶対しない	多分しない	迷う	多分する	受講する	未回答
39	72	191	239	164	9
5.5%	10.1%	26.8%	33.5%	23.0%	1.3%

n=714

学生の反応

最終回 #15 の activity-report 提出時に、科目全体に関しての主観評価を問う多選択肢式アンケートをとっている。2018 年度の結果を表-3～8 に示す。全体として、内容は難しめで負担も大き目だったが、多くのことを学んだと考え、必修でなかったとしてもこの科目を取りたいと考える学生が多くいたという結果が得られた。

そのほか、レポート類の自由コメントを見ても「ハードだったが勉強になった」「楽しかった」という前向きなコメントが多く見られた。「学期中は忙しかったので夏休みに復習したい」のようなコメントも複数見られた。個別の内容については「CUI が使えるようになった」「LaTeX が使えるようになった」「HTML が書けるようになった」というものが多く見られた。これらのコメントから見ても、学生は本科目の意図するところは理解し、十分受け止めていると思われる。

その一方で「コンピュータが好きな学生ばかりではない」「3 類（非情報系）に進むのでこのような内容はやりたくない」のような後ろ向きなコメントも少数ではあるが見られる。このような学生にも理解を得られるためにどうしたらいいかは、今後の課題だと考えている。

参考文献

- 1) 高橋尚子：国内 750 大学の調査から見てきた情報学教育の現状—(3)一般情報教育編—, 情報処理, Vol.58, No.6, pp.526-530 (June 2017).
- 2) 久野 靖, 江木啓訓, 赤澤紀子, 竹内純人, 笹倉理子, 木本真紀子：コンピュータサイエンス入門教育の題材としてのアセンブリ言語プログラミング, 情報処理学会論文誌教育とコンピュータ, Vol.4, No.2, pp.23-36 (June 2018).

(2018 年 8 月 3 日受付)

久野 靖 (正会員) y-kuno@uec.ac.jp

1984 年東京工業大学理工学研究科情報科学専攻博士後期課程単位取得退学。同大学助手、筑波大学講師、助教授、教授を経て現在、電気通信大学情報理工学研究科教授。筑波大学名誉教授。理学博士。プログラミング言語、プログラミング教育、情報教育に関心を持つ。