べた語義

Vol. 166

CONTENTS

基般

COLUMN

未来の学びを創る
〜探究と協働で育む，学習者主体の授業デザイン〜

現行学習指導要領が高校でも実施されるようになって 3 年目が終わろうとする 2024 年 12 月，文部科学大臣は，中央教育審議会へ，学習指導要領改訂に向けた諮問を行いました．2025 年度は，諮問で「道半ば」とされている現行学習指導要領の趣旨を一層活かした教育活動を行うことが求められます．学習指導要領前文にある「持続可能な社会の創り手となる」ために必要な資質・能力を育むには，生徒を主語とした教育を推進することが重要です．特に，主体的・対話的で深い学びを促進し，個別最適な学びと協働的な学びを一体的に充実させることが求められています．

2016 年 12 月の中教審答申では「まず学習する子供の視点に立ち，教育課程全体や各教科等の学びを通じて『何ができるようになるのか』という観点から，育成を目指す資質・能力を整理する必要がある．その上で，整理された資質・能力を育成するために『何を学ぶか』という，必要な指導内容等を検討し，その内容を『どのように学ぶか』という，子供たちの具体的な学びの姿を考えながら構成していく必要がある」と述べています．子供が「どのように学ぶか」の姿として示されたのが「主体的・対話的で深い学び」で，主語は「子供」です．子供の「主体的・対話的で深い学び」を実現するための授業実践を，私もこれまで積み重ねてきました．

例として，「情報 I」の単元「ネットワーク構築」における授業実践を紹介します．まず，情報通信ネットワークの仕組みを学ぶために，生徒たちはグループでネットワークの知識をクラウド上に協働で整理し，ネットワーク構成図を作成しました．そして，自分たちの計画に基づいて，ネットワーク機器の接続や設定を行い，実際にネットワークを構築しました．この過程で，生徒たちはインターネット上の情報や生成 AI を活用し，必要な情報を取得しながら試行錯誤を重ねていました．彼らは主体的に学び，対話を通して思考を深め，実践を通じて理解を深めていました．

このような実践は，デジタル学習基盤を前提とした新たな時代にふさわしい学びの形を示しています．生徒は，「習得・活用・探究」という学びの過程の中で，各教科等の特質に応じた「見方・考え方」を働かせながら，知識を相互に関連付けてより深く理解したり，情報を精査して考えを形成したり，問題を見いだして解決策を考えたり，思いや考えを基に創造したりすることに向かう「深い学び」が実現しています．生徒が自律的に学び，学びの意義を見いだすことにより，生涯にわたって主体的に学び続ける力を身に付けることができます．

社会全体の生産性や創造性を高めるためには，デジタル人材の育成が不可欠です．そして，異なる価値観を持つ多様な他者と対話し，問題を発見・解決できる「持続可能な社会の創り手」を育てることが，今後の教育に求められています．

すべての子供が多様で豊かな可能性を開花できるよう，学校や教師は柔軟に変化し続け，より良い学びを追求していく必要があります．そのためには，高度な問題解決を情報および情報技術を活用して行い，新たな価値を創造できるように，教師が体験的な学びを実現する授業を設計・実践していくことが重要です．教師が，生徒が主体的に学び続ける環境を構築し，未来の社会で必要な力を培い，自らの意志で生きていけるよう支援することは，今後の教育における重要な課題です．その実現に向けて皆様とともに取り組みたいと思います．

須藤祥代（国立教育政策研究所教育課程研究センター／文部科学省初等中等教育局）（正会員）s-sudo@nier.go.jp

国立教育政策研究所教育課程研究センター 研究開発部 教育課程調査官．文部科学省初等中等教育局 学校情報基盤・教材課／教育課程課 情報教育振興室 教科調査官，参事官（高等学校担当）付 産業教育振興室 教科調査官．

LOGOTYPE DESIGN...Megumi Nakata

ARTICLE

大学における一般情報教育の 30 年

中西通雄
追手門学院大学

一般情報教育とは

一般情報教育とは，専攻する分野に関係なくすべての学生が学ぶ教養教育としての情報教育を指す．本会誌を含めていくつか解説されているが（文献 1），文献 2）など），本稿では筆者がかかわってきた経験をもとにして，その歴史を振り返り，今後への課題について述べる．

❑ 情報教育のはじまり

日本では，1970 年に情報工学科や情報科学科が誕生し，情報の専門教科としての知識体系や教育方法が固められてきた（図 -1）．まだ一般情報教育のなかった時代の 1988 年には，情報処理学会に「コンピュータと教育研究会」（CE 研究会）が発足している．また同年から情報処理教育研究集会が国立大学情報処理教育センター協議会と文部省（現文部科学省）との共催で開催され，これが現在の大学 ICT 推進協議会（AXIES）の年次大会につながっている．1992 年には私立大学情報教育協会が設立され，学部での教育を含むさまざまな研究発表がされてきた．1993 年からは，CIEC（コンピュータ利用教育学会）の PC カンファレンスが開催されている（最初の 3 年間は全国大学生活協同組合連合会による開催であった）．

つまり，情報教育は 1990 年代前半にスタートしたと言えよう．これは，1991 年にインターネットが商用利用できるようになったことに続いて，GUI の Web ブラウザが開発され（1993 年に Mosaic，1994 年に Netscape Navigator，1995 年に Internet Explorer と OmniWeb），インターネット利用の世界が一気に開花したこと，および企業等でのパソコンの利用の台頭によるものである．

この流れの中で，教養教育としての一般情報教育はどのような内容で始められたかを振り返ってみよう．

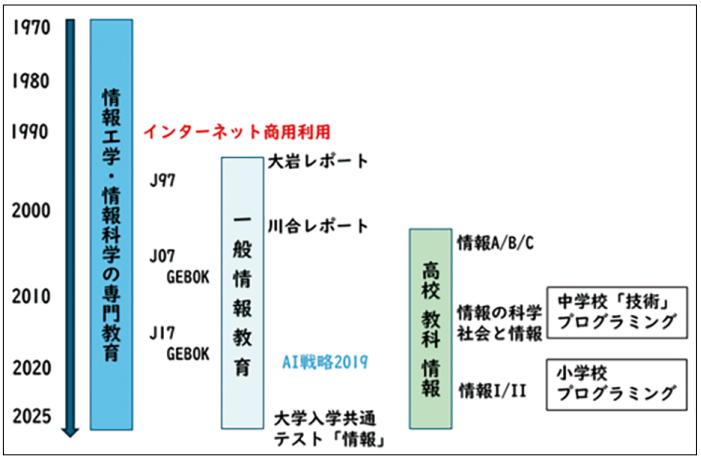

図 -1　情報教育の 30 年

❑ 大岩レポート（1992）

情報処理学会では，文部省から委嘱を受けて，「一般情報処理教育の実態に関する調査研究」の調査報告書を 1992 年に作成した．委員長（慶應義塾大学大岩元教授）の名前をとってこれを大岩レポートと呼ぶ．以下，筆者が文献 3）にまとめたものを引用する．

大岩レポートでは，一般情報処理教育の目的を「計算機並びに情報という概念を理解させ，自在に活用できるようにすること」と定め，さらに，具体的な教育内容として次の 3 つを挙げて，これらをバランス良く教えることが重要であると述べている．

(1) コンピュータリテラシー教育

ワープロや電子メールといった道具を，単な

る技能としてではなく，その概念，動作原理を含めて正しく利用できるようにする教育.

(2)「プログラミング」教育

特定のプログラミング言語の習得だけを目的とするものではなく，問題を発見して，それを解決するシステムを創り出し，さらにできあがったシステムの使用を通じて新たな問題を発見するという，システム進化の過程全体の教育.

(3) 情報科学教育

情報科学の世界観，面白さ，深さを伝えていくような教養主義的教育.

大岩レポートは，多くの大学で受け入れられ，これに基づいて一般情報処理教育が行われるようになった．しかしながら多くの大学では，それに割り当てることができる授業コマ数の制約等のため，(1) のコンピュータリテラシー教育のみが取り上げられる傾向にあった．さらに，ワープロや表計算ソフト等のアプリケーションプログラムの使い方を教えることがコンピュータリテラシー教育である，といった誤解も少なからずあった.

❑ 大阪大学での一般情報処理教育

筆者が当時所属していた大阪大学においては，教養部解体に合わせて 1994 年度から「情報活用基礎」という 2 単位科目が新規開講され，6 学部で必修科目，1 学部で選択（1995 年度から必修）科目として，情報処理教育センターの教室に導入された 400 台の NeXT workstation を用いた教育が開始された．筆者がかかわった学部の授業内容は，大岩レポートの (1) コンピュータリテラシーを中心とするものであり，コンピュータの仕組み，日本語入力の仕組み，メール配送の仕組みなども，作成した教科書『NEXTSTEP によるコンピュータ・リテラシー入門』（アスキー，1996）に含めていた．また，(2)「プログラミング」教育についてはこの科目 (90 分×15 回) 中に適切に組み入れることは難しかったため，LaTeX による文書作成や HTML による Web ページ作成で代替していた.

大阪大学の教育用コンピュータシステムは，授業での利用が急増して実利用者総数は約 1 万 2 千人となり，機種更新ではサーバ群と 800 台近くの利用者用コンピュータの規模で構築することになった．1995 年末にはパソコン用 OS として Windows 95 が発売されていたが，利用者用コンピュータは NeXT workstation から Intel PC に変え，OS には NEXTSTEP を継続して採用した．さらに 4 年後の機種更新でも利用者用コンピュータに Intel PC と Vine Linux を採用した．また，2000 年には，サイバーメディアセンター (現 D3 センター) の誕生に合わせて，選択科目として「コンピュータのしくみ」,「情報科学入門」,「情報社会と倫理」,「計算機シミュレーション」,「情報探索入門」の 5 つを選択科目として新設した.

❑ 川合レポート (2002)

情報処理学会は，2000 年度に文部省から「大学等における一般情報教育の在り方に関する調査研究」の委嘱を受けた．委嘱の背景には，1999 年 3 月に文部省が高等学校の新しい学習指導要領を告示したことが挙げられる．つまり，2003 年度から高校で普通教科情報として，「情報 A」,「情報 B」および「情報 C」のうちから 1 科目が必履修となることが決まっていた.

調査研究の委員会では，2006 年度に新しい学習指導要領に基づいて学習した生徒が大学に入学してくることも踏まえて，標準カリキュラムについて検討された．2002 年に発行された報告書「大学等における一般情報処理教育の在り方に関する調査研究」[4] を，委員長 (東京大学川合慧教授) の名前をとって川合レポートと呼ぶ.

川合レポートでは，大学での一般情報処理教育の実施状況の調査結果から，コンピュータの操作スキル教育が中心とされていて基礎的な原理や社会とのかかわりに関する内容が十分に扱われていないケースが多いことが明確にされた．そして，一般情報処

理教育カリキュラムとして，2つの2単位科目「情報とコンピューティング」と「情報コミュニケーション」を必修科目とし，5つの補完的科目あるいは選択科目として開講することを推奨している．

2004年には，2つの必修科目の科目名を書名とする教科書がIT Textシリーズとしてオーム社から発刊された．

❑ 高校で情報が始まったが

高校では，2003年度から教科情報がスタートしたが，実態としてはほとんどの高校で「情報A」だけが開講された．日経コンピュータ2005年4月4日号（p.124～130）には，高校の情報の教育について，「実態は『町のパソコン教室』以下」と揶揄する記事が掲載された．ワープロ・表計算ソフトの操作スキル教育に終始しているという訳である．高校現場での情報教育がこのように評されたことの背景には諸般の事情があったが，本稿では省略する．結果として，2006年度からも大学での一般情報処理教育の内容は変更する必要がなかった．

❑ カリキュラム標準 GEBOK

1997年に情報処理学会は「大学の理工系学部情報系学科のためのコンピュータサイエンス教育カリキュラムJ97」を発表した．このカリキュラムは，米国のComputing Curriculaを参考として作成された．また，一般情報教育は対象外であった．

その後，2010年には，「大学の理工系学部情報系学科のためのコンピュータサイエンス教育カリキュラム標準J07」が発表された[5]．このJ07に一般情報処理教育の教育理念，カリキュラム，教育体制と設備などが含められたことは，情報処理学会でコンピュータサイエンスを一般情報処理教育の親学問として位置づけたと言えよう．J07の中の一般情報処理教育のカリキュラム標準は，川合レポートに提案された内容が踏襲され，さらに一般情報処理教育の知識体系（GEBOK：General Education Body of Knowledge）が新たに策定されて教育時間数も例示された．詳細は文献6）に解説されているのでここでは省略する．また，プログラミングに関しては次章で述べる．

カリキュラム標準は，さらに更新されて2018年にJ17が発表されている．J17の中での一般情報教育の知識体系は，J07での内容を踏襲しつつ，新たに2つのエリアが追加されたほか，情報倫理エリアの教育時間数も増やされている[7]．

なお，情報処理教育研究集会は，2006年度からその名称が「処理」の2文字を削除した情報教育研究集会に変わった．情報処理学会でも2007年度から一般情報処理教育の名称を一般情報教育に変更されている．ちょうどこの頃，日本で「情報フルーエンシー」という言葉が使われるようになったことも記憶しておきたい．

プログラミング教育

大岩レポートの（2）は，括弧つきの「プログラミング」であり，言語習得だけが目的ではない．しかし，20行程度のプログラムを書けるようになることは，将来自分でプログラムを書くことがないとしても，問題解決のための素養として必要であろう．

❑ GEBOKでのプログラミング教育は5コマ程度

GEBOKで必須とされた「ALP1 アルゴリズムとプログラミング」は実際のプログラミング演習による体験を含めることが要件とされている．J07でもJ17でも，その学習時間として90分×5回程度が想定されている．また，文献6）のp.772では，「アルゴリズムの記述として大学入試センターの『情報関係基礎』で使われているDNCL試験用手順記述言語を用いることも考えられる」とされている．

❑ プログラミング初学者教育の実施例

プログラミングの演習ではCやJavaなどが用い

られてきて，現在では Python もよく用いられている．日本のプログラミング初学者が躓く要因の 1 つは英語である．ソースコードの while や for などは受け入れても，英文で書かれたエラーメッセージは，そこに用いられる単語の意味が理解できないために読まない．漢字を用いて生活しているところに，アルファベットを持ち込まれることによる認知的・心理的な抵抗感もあるようである．それ以外にも，初めて接する条件分岐や繰り返しのブロック構造にとまどう．

筆者は，1995 年から 2022 年まで大阪大学人間科学部の一般情報教育を担当してきた．文学部でもほぼ同じ内容で，両学部ともプログラミングには，2006 年頃から 90 分× 5 回程度をあてた．具体的には，初学者に焦点をあてて，日本語で命令を表記するプログラミング環境 PEN を用いてきた．PEN は，DNCL と似た文法で記述したプログラムを実行できるほか，プログラムの入力支援機能があり，ボタンをクリックすることで繰り返し構造や条件分岐の構造を簡単に入力できるので，タイピングのエラーは生じにくい特長がある．

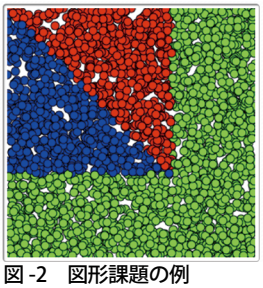
図 -2　図形課題の例

大学 1 年生で初めてプログラミングを学ぶ学生が大多数であったので，2019 年頃からはプログラミングの最初の回で Scratch を学習し，そのあとブラウザ上で使える wPEN を使用している．プログラムは，短冊（プログラムの 1 行）を組み立てて作成でき，ブロック範囲を示す色が付くので構造も分かりやすい．wPEN を用いた演習の最後では，図 -2 の図形を描くプログラムを作るような課題を与えている（図 -3 のプログラム）．

図 -3　wPEN でのプログラム例

2022 年からの 3 年間は，私立大学の経営学部情報システム専攻 1 年生のプログラミングも担当してきた．大学入学までのプログラミング経験は着実に上昇しているが，2024 年度でも 3 割強の学生が未経験であった（表 -1）．

この授業では，105 分× 全 13 回の授業の最初の 2 回で Scratch を用いてプログラミングの感触をつかんでもらい，続く 5 回で wPEN によるプログラミングを実施し，残り 6 回で Python の基礎を教えた．3 年間の受講後のアンケートでは，この Scratch → wPEN → Python の順で学ぶことが毎年圧倒的に支持されている．この知見は高校でのプログラミング教育に役に立つと考えている．今後高校でのプログラミング教育がさらに進むと，大学での内容も教え方も変化するであろう．

❑ プログラミングが苦手なのはなぜ?

筆者は 3 つの大学でプログラミング教育を経験したが，3 カ所とも 1 ～ 2 割程度の学生がついて来られない状況があった．初回に Scratch に触れてもらうようにしたのも，すべての学生がプログラミングの授業についてきてほしかったからである．一部の学生がいわゆる落ちこぼれ状態になる傾向は，海外のプログラミング初学者教育でも生じており，筆者の知る限り，まだ解決の決定打はなさそうである．ある高校では，

表 -1　プログラミング経験

	2022	2023	2024
いままでに作ったことがない	74%	44%	37%
初めて作ったのは小学生のとき	3%	9%	8%
初めて作ったのは中学生のとき	4%	24%	20%
初めて作ったのは高校生のとき	19%	24%	35%
有効回答数	73	68	60

2025年1月の大学入学共通テストのプログラミングの試験問題の成績が，英語の模擬テストの成績と高い正の相関があったことが報告されていて興味深い[8]．課題文を読解する力と論理的に組み立てる数学的な力の両方が関係するはずで，プログラミング教育には認知科学に基づくアプローチでの研究も必要であろう．

これからの一般情報教育

GEBOKの新しい版を作る時期になっている．大学入学共通テスト「情報」の受験の有無により，知識・技能の差が拡大していると考えられるが，ここでは高校の「情報I」を基礎として大学で積み上げる内容として，筆者が日頃考えていることをいくつか挙げてまとめとしたい．

❑ ワープロ操作スキル

大学では，A4判で数ページ程度の分量のレポートが課されることもある．特に卒業論文では，多くの図表や参考文献を含めるので，その採番と相互参照，図表目次の作成などにワープロを使いこなす必要がある．数系・情報・物理などの学科ではLaTeXを使わせるケースもあるが，一般にはWordがデファクトスタンダードになっているので，知的生産の道具として低学年のうちにWordの操作を習得させておくのがよい．

❑ GEBOK内容の見直しに向けて

コンピュータの操作，たとえばファイルの作成や削除の操作によってOSのファイル管理が何をしているかを，クラウドとの関連を含めて理解することはセキュリティ面でも重要である．また，コンピュータを構成する論理回路自体よりも，「産業のコメ」としての半導体とは何か，その製造体制の現状，さらにはデータセンターやクラウドのビジネスなども含めたい．

ソフトウェア関連では，プログラミング演習だけではなく，要求仕様・外部仕様を書く演習やUMLのユースケース図などを利用して仕様を整理するスキルはAI時代に必要であろう．また，大岩レポートの(3)に関連して，Web画面上にパーソナライズされた広告が表示されるときに，どういう仕組みで複数社の広告が競争を勝ち抜いて表示されるのか，あるいはタクシーの配車アプリや宅配便の追跡システムを紐解いて議論するのもよい．

セキュリティ関連では，Cookieと個人情報保護，プライバシー，ブラウザへのプラグインの功罪，パスキーなどの新しい認証技術，クラウドのセキュリティなど，仕組みも含めた理解が重要である．

情報倫理関連では，エコー・チェンバー／フィルターバブルが生じる構造や，知的財産として商標，特許，使用許諾などの概念だけでなく具体的事例を含めた学びも欠かせない．

本稿では，一般情報教育の誕生から現在に至るまでの経緯を概説した．今後の教育内容を検討される際の参考になれば幸いである．

参考文献

1) 河村一樹：大学における一般情報（処理）教育，メディア教育研究，6(2)，メディア教育開発センター（2010）．
2) 玉井哲雄：東京大学における一般情報教育，ぺた語義，情報処理，Vol.52, No.10 (Oct. 2011).
3) 中西通雄，松浦敏雄：情報処理教育の2006年問題への対応，サイバーメディアフォーラム（2005.9），https://doi.org/10.18910/73023
4) 情報処理学会：大学等における一般情報処理教育の在り方に関する調査研究（2002.3），https://www.ipsj.or.jp/annai/committee/education/report3c.pdf
5) 情報処理学会：情報専門学科におけるカリキュラム標準J07（2010.6），https://www.ipsj.or.jp/12kyoiku/J07/J0720090407.html
6) 河村一樹：一般情報教育（J07-GE），情報処理，Vol.49, No.7 (July 2008).
7) 情報処理学会：カリキュラム標準一般情報教育(GE)（2018），https://www.ipsj.or.jp/annai/committee/education/j07/ed_j17-GE.html
8) 藤岡健史，中村央志：共通テスト「情報I」の高1実施結果報告―試作問題との比較から見えるもの―，情報科教育学会近畿北陸支部研究会講演口頭発表(2025.3)．

(2025年4月6日受付)

中西通雄（正会員） Nakanishi.Michio@gmail.com

大阪大学基礎工学部情報工学科卒業．三菱電機を経て，1990年に大阪大学，その後大阪工業大学，追手門学院大学で勤務．2025年4月よりフリー．追手門学院大学ベンチャービジネス研究所研究員，本会一般情報教育委員会委員．

コンピュータサイエンスの大衆化 ―教科「情報」に CS の「知恵」を―

久野　靖
電気通信大学

教科「情報」と情報入試のインパクト

高等学校の教科「情報」は，2003 年に選択必履修の形で始まった．筆者をはじめ情報教育に関心を持つ研究者は，これで「すべての高校生が情報を学ぶ」のだから，世の中は大きく変わる（つまり情報，ひいてはコンピュータのことをみんなが知っている世の中になる）と期待した．

しかし，そうはならなかった．大学入試にほとんど関係せず，単位数も少ないことから，「情報」は日陰の存在で在り続けた．教える教員も非常勤や「情報」の免許を持たない教員への臨時免許等の付与が多かった．そしてコンピュータサイエンス（CS）について言えば，選択される情報科目のうち，プログラミングをはじめ CS に関係の深い内容を含む科目の履修人数は 10 ～ 20% であり，その選択は生徒ではなく学校によってなされた．

ただ，CS 以外の内容で，目に見える変化もあった．それは「著作権」で，文部科学省がその教育に力を入れ，どの情報科目選択でも教えられるようになっており，また専門性が高くない先生であっても扱いやすかったためか，高校卒業生はだれでも著作権とその遵守について知っていて，その点では社会人より優っている，という状況にある☆1．

そうこうするうちに，2022 年から情報科は「情報 I」が（選択なしに）必履修となり，すべての生徒がプログラミングを学ぶこととなった．そして文部科学省の指導もあって，情報の免許を持つ教員が情報科を教えるという「当たり前の」ことが，ようやくほぼ実現しつつある．

これに合わせて，2025 年大学入試から「情報」が大学入学共通テストに加わり（科目は「情報 I」），国立大学受験生は必ずこれを受験することとなった．2025 年の「情報 I」の受験人数は 30 万人超で，共通テスト受験者の 60% 強であった．

これらにより，CS に関係の深い内容についても，著作権と同様，「高校卒業生は誰でも知っている」になるのでは…… このように「必履修」と「情報入試」の情報科に与えるインパクトは大きく，これによって 2003 年当時の「期待」を再度押入から取り出している，というのが筆者の現況である．しかし虫食いやカビは大丈夫だろうか…….

「情報 I」とコンピュータサイエンス

コンピュータサイエンス（CS）とは，コンピュータの仕組みやデータ処理，アルゴリズム，プログラミングなど，情報技術に関連する幅広い分野を研究する学問を指し，情報科学や情報工学とも呼ばれる．より広く，情報の原理や情報社会（いずれも文系）等まで含めた場合は情報学[1]と呼ばれるが，本稿では理系の CS の範囲を考える☆2．

「情報 I」の内容範囲と CS とのかかわりを図 -1 に示した．こうして見ると，およそ半分の内容が CS に関係しているといえる．「情報デザインと効果的なコミュニケーション」については，情報デザイン，特にユーザインタフェースが CS の内容となる．「モデル化，シミュレーションとモデル評価改善」については，シミュレーションの部分，「データの蓄積管理と

☆1　情報科の始まりから 20 年が経過した今日では，「若手は知っているが，シニアはあまり知らない」状況になっている．

☆2　今日では CS に「社会的視点と情報倫理」など文系的なものも含めるようになっている．

表現，データの収集整理分析」についてはデータベースやデータの表現の部分が CS の内容となる．

コンピュータサイエンスで学ぶこと

ここで「期待」の中身を見直してみる．「みんなが情報のことを知っている」というのは，たとえばプログラミングであれば，誰もがコードを書いたことがあり，相当数がそれを得意とする，つまり必要ならコードを書いて問題を解決できる，ということになるだろうか．

これは，我々コンピュータ屋（大学等で CS を学んだ者）には「当たり前」だったが，それが「我が国の全国民」になるとすれば（時間はかかるにせよ），画期的な変化だと言ってよい．まさにコンピュータサイエンスの大衆化である……，と喜んでばかりはいられない．

というのは，我々が CS で学んでいることは決して「プログラムが書ける」「データベースにデータが保管できる」「ネットワークが使える」こと**だけ**ではないからである．

我々は「〜できる」に加えて，さまざまな「知っておくべきこと（知恵）」を身に付けさせられ，その結果その道の専門家になっている．それ自体はこれからも続くだろう（図 -2 左）．

一方，「大衆化」によって「〜できる」を学んだ多数の人は，専門家までは必要としない範囲の CS のあれこれを担うようになると期待される．しかしその人たちは，「知っておくべきこと」を学ぶ機会はない，というのが現状である（図 -2 右）．

（1）情報社会の問題解決
　情報と情報技術を活用した問題発見・解決
　情報と法・制度，セキュリティ，責任とモラル
　情報と情報技術の活用，望ましい情報社会構築
（2）コミュニケーションと情報デザイン
　メディアとコミュニケーション手段の特徴
　情報デザインが人や社会に果たす役割
　情報デザインと効果的なコミュニケーション☆
（3）コンピュータとプログラミング
　ハードの仕組み・特徴，情報表現と計算の限界★
　アルゴリズムとプログラミングによる活用★
　モデル化，シミュレーションとモデル評価改善☆
（4）情報通信ネットワークとデータの活用
　ネットの仕組み，プロトコル，セキュリティ★
　データ管理，ネットと情報システムのサービス★
　データの蓄積管理と表現，データの収集整理分析☆

★—CS が中心となる，☆—CS と一部関係

図 -1 「情報 I」の内容と CS

そうでなく，「〜できる」とともに「CS の知っておくべきこと」を，大衆化に応じたレベルで身につけてもらう必要はないだろうか．つまり「情報科で全員が学ぶ内容」[☆3] に，そのような内容を加えるような活動をするべきではないだろうか．それがあるべき「コンピュータサイエンスの大衆化（Popularization of Computer Sciense, PCS）」であろう．これが現在筆者が思うことである[☆4]．

プログラミング：CS で学ぶこと

いちばん分かりやすいのは「プログラミング」なので，これから取り上げよう．プログラミングは「書ける」ようになるまで大変なので，とかく「〜できる」方向だけに注力されがちである．しかしすこし考えてみれば，我々は「書ける」に加えて次のことも知っておくことが大切だと分かっている．

- プログラムは「何をするかが明確で」「それを正しく行う」ように作るべきだ．きちんと考えずにプログラムを作るとこれらが毀損しやすいが，速く動き便利に見えても，正しくないプログラムは無価値（むしろ有害）である．
- プログラムは速さや小ささよりまず「分かりやすく」書くべきだ．そのほうが間違えにくく（間違ったコードは速くても無価値），直しやすく（大半のコードは間違いや改修のため直される），誰にも読みやすい（コードは他人が直すことも多いし，1 週間後の自分は他人[☆5]）．

☆3 現時点では必履修科目「情報 I」2 単位がこれに相当するが，将来的には時間数や科目が強化されてほしいと思う．

☆4 本会情報入試委員会での議論では，CS を学ぶとは「作る人」を目指すことであるが，それは行きすぎで，「よく分かっている，使う人」を目指すのでいいのではないかという意見もあった．

☆5 コードは書いたときは当然分かっているが，1 週間もすると細かいところを忘れてきてしまうことを指して「1 週間後の自分は他人」という言い方をする．

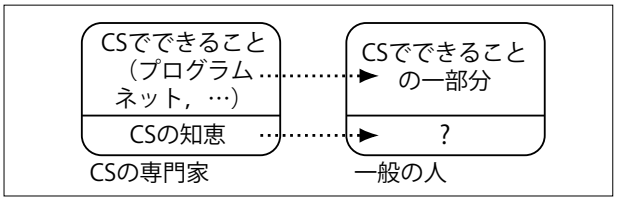

図 -2 コンピュータサイエンスの大衆化

- プログラムはきれいに「構造化」して（関数に分けたりオブジェクトを構成したりして）書くべきだ．そのほうが 1 つずつの単位が小さくて分かりやすいし，単位の外側からは内部の詳細を気にせず「抽象化」して扱える．
- プログラムには適切にコメントをつけ，どの関数は何をするか，どのデータ構造はどう使われるかなどが分かるようにする（コメントで不足ならドキュメントを書く）．そうしなければ，そのコードは苦労して読み解かなければ手がいれられず，使い続けられない．
- プログラムは適切にテストされるべきだ．コードはバグを含むのが普通であり，テストなしには信用できない．テストは保管され，コードを直したときは再実行されることで，信用を維持しなければならない．

CS の教えはまだまだたくさんあるが，大衆化のためということなら，この 5 つくらいが妥当そうである．そして，これらを知らないか無視していると必ず痛い目に遭うことも，我々はよく知っている．

一方で，「自力で」これらをクリアする人は多くなさそうである．ということで，これらは「全員が学ぶ内容」に入れてほしい．

ソフトウェア工学：CS で学ぶこと

前章では「プログラムを作る」と言ってきたが，自分以外の人が使うものを，かつ／または複数人数で開発する，つまり「ソフトウェア開発」の場合は，「作れる」に加えて，またさらにその方面の CS の教えを知っておくことが大切である．

- ソフトウェアはやみくもに作りはじめるのでなく，まず要件・要求をまとめ（＝利害関係者やその求めるものをはっきりさせ，どのように問題を解決するか決める），要求仕様を定める（どのようなソフトで，何と何が機能として含まれるか決める）．これらが明確でないと，誰も望まないものができてしまう恐れがある．
- ソフトウェア開発でコードを書く場合の生産性は単に「プログラムを書く」場合よりずっと小さくなる．これは文書化やテストに手間が取られることもあるが，複数人で 1 つのものを作ることの難しさによることが大きい☆6．
- ソフトウェア開発では開発プロセス（開発を進める枠組み）を決め，それに基づき進捗を管理する．分かりやすいのはウォーターフォール型☆7 だが，後になって仕様を変更するというトラブルが付きものである．これに対し，反復型のプロセス☆8 は仕様変更の問題は少ないが，運用に難しさがある．

ソフトウェア工学に関する CS の教えは，大衆化により，自分たちが開発者となるため必要になる面もあるが，開発者と協力してソフトを作る（ユーザサイド）場合にも大切である．

データベース：CS で学ぶこと

データベースは情報システムの中核であり，重要なものだということはよく知られている．しかし，現在の「情報 I」の教科書中では，データベース自体に割かれるページ数は多くなく，関係データモデルと DBMS の主要な機能について一通り述べられている程度であることが普通である．

一方，我々はデータベースについて，「データベースにデータが格納でき，問い合わせできる」ことに加え，次のことを知っているべきだと考える．

- データベースは「データ独立☆9」を実現すべきであり，これによって複数のプログラムが共通の

☆6 このため，早く開発しようと人を増やすと，逆にスケジュールは遅れること（ブルックスの法則）が知られている．
☆7 要求仕様・設計・制作・テストのように段階を経て開発を進め，後戻りを許さないプロセス．
☆8 何回も同じ段階を（らせん状に）繰り返しながら開発を進めてゆくもので，アジャイル型と呼ばれるものが代表的．
☆9 データとプログラムが互いに独立で，性能や機能のためにデータの形を変更してもプログラムの修正が不要なこと．たとえば，あるテーブルに新しい機能のためフィールドを追加しても，その機能に関係しないプログラムは修正不要なこと．

データにアクセスできる☆10.

- データベースは「データの一貫性☆11」が重要であり，一番避けるべきことは，データの損失や破損である．このためにさまざまな制約や不便さを甘受している．

これらのうち1番目については，今日では情報システムはDBMS☆12にデータを保管するのが当り前であり，システムの各所からその必要なデータにアクセスするように自然に作られるので，何かを変えるという必要性は大きくない．

一方2番目は，データ操作の記述が正しく，かつDBMSが普通に動いていれば達成されているはずだが，その原理や裏方仕事も含めて必要なことを知っていて分担しなければ，競合や事故によりデータの破損などの憂き目に遭いかねない．

「情報I」の教科書で，並行制御・障害対策・安全対策について説明しているものも複数あるが，並行制御についてはさらに次のことが必須である．

- 1カ所で複数のデータ操作を行う場合は，必ずトランザクションとして実行し☆13，原子性を持たせロールバックに備える．

これは，データベースのデータ操作を記述する者は必ず守るべきである☆14．

また障害対策について次のことが必要である．

- データベースのデータは管理者がスケジュールに従ってバックアップし，障害で必要な際はここからデータが復元される必要がある．
- 故障や電源喪失などの際にはDBMSが働いて最後の整合性のある状態が復元されるが，それがどの時点でそこから後の操作は必要なら再実行，などの判断は管理者の仕事になる．

これらは，データベースが使われるシステムを運用する者が必ず理解し，実行すべきである．

☆10 データ独立でないと，機能を増やすごとに既存のプログラムを修正するのが繁雑となり，新機能は共通データでなく別の新規データを使う方法で作りがちになる．

☆11 データが意図されたとおりに記録され取り出せること．

☆12 データベース管理システム（Data Base Management System）．データベースを実現するソフトウェア．

☆13 たとえばSQLなら複数の操作を`BEGIN`と`COMMIT`で囲む．

☆14 これを行わなくても負荷が軽いときは何事もなく動くが，負荷が大きくなると知らないうちにデータが破損する結果を招きがちである．

ネットワーク：CSで学ぶこと

ネットワーク（インターネット）は，我々の日常に深く根づいており，その使い方は言われるまでもなく分かっているという人が多いと思われる．また，現在の「情報I」でもネットワークの仕組みや構成要素，プロトコル，セキュリティについて扱うこととなっていて，ここまでに挙げたほかの分野のように大きく足りない部分はなさそうである．

それでも，「繋がらない」トラブルは，ネットワークに関する総合的な知識が必要で，簡単ではない．ただ，「繋がらない」害は目に見えるものであり，大きな危険にはつながりにくい．

どちらかというと，ネットワークが「使える」ことに加え，セキュリティの脅威に対する考え方を中心に，次のCSの教えを知っておきたい．

- ネットワークではEnd-to-End（E2E）の原則☆15が多くの場面で適用され，エラー処理やセキュリティ☆16など，これでしか解決できない問題も多い．一方で性能などE2Eだけでは対応しにくい問題もある．
- セキュリティを保った通信には暗号技術が使われるが，その際に「秘匿すべき情報を公開しない」「時とともに暗号技術の安全性は脅かされるので，時代に合ったものに乗り換えていく」などを原理とともに理解し，正しく使うことが不可欠である．

1番目については，セキュリティに関して，E2Eでない暗号化は危険ということである．

たとえば無線LANで安全な暗号をと言われるが，それはその無線LANで完結する通信（無線ルータの設定や無線接続プリンタへの印刷など）にのみ有用で，

☆15 信頼性やセキュリティなどアプリケーション固有のニーズに従う機能は，ネットワークではなく通信し合う終端ノードの機能として実現すること．

☆16 たとえば通信線ごとに暗号化する方法だと，通信線上のデータは安全でも，中継点で一度復号し次の通信線のため再び暗号化するので，中継点に対する攻撃には無力である．一方，通信の始点で暗号化し，終点まで復号しなければ（E2E），途中のどこを攻撃されても解読の心配はない．

より遠くへの通信は TLS（https）や ssh などの E2E 暗号化通信を使うべきである．

2 番目については，暗号技術の原理は現在の「情報 I」の教科書でも扱われているが，それが自分の行動とどう関係するのかという点では心許ない．たとえば公開鍵と秘密鍵の対をファイルに生成した後，公開鍵は必要に応じてコピーするが，秘密鍵は持ち出さず，他人に見られないよう保護しなければならない．また，その際に選ぶ暗号の適切な方式や鍵の長さは，計算機の速度向上とともに変わっていくことに注意を払わなければならない．

計算論的思考との違い

多くの人に CS のエッセンスを学んでもらう，というと「計算論的思考（Computational Thinking, CT）」[2] が思い浮かぶ．CT は「コンピュータサイエンティストのように考える」とも言われ，多くの人が手順的・プログラミング的思考と，さらには抽象化，仮想化，メタ認知など，より広い CS の思考法を身に付け，より良く問題解決でき，CS に親しむことをめざす考え方である（図 -3 左）．それはもちろん，素晴らしいことだと思う．

一方，本稿で取り上げているのは，プログラミング，データベース，ネットワークなどが広く世の中に行き渡りつつある中で，これらをキャッチアップしようとする人たちのために何をすべきか，である．つまり，そのような人たちが，これらを長く扱ってきた CS の中にある「教え」を知らないがために，先人の落ちた「落とし穴」に落ちそうになるのを回避させなければ，と考えるわけである．

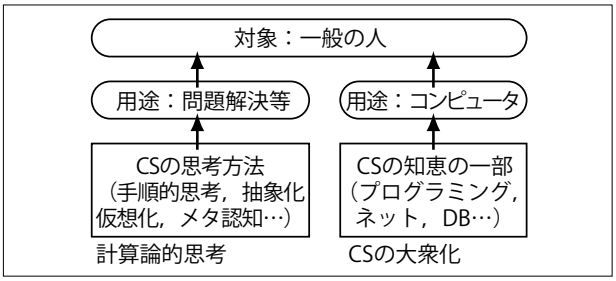

図 -3　計算論的指向と CS の大衆化の違い

そのための運動を「コンピュータサイエンスの大衆化」と名付けているわけだが，それは直接コンピュータを扱うことが前提になるので（図 -3 右），その点が計算論的指向とは別のものと言える．

コンピュータサイエンスの大衆化の今後

それでは「コンピュータサイエンスの大衆化」運動はこれから何をすべきだろう．まず，ここで挙げたものが本当にふさわしいか，時間をかけて学んでもらう価値があるか，精選していく必要がある．また，ほかに追加するものの検討もしたい，

そして，これらをどのように学んでもらうかも，考えていく必要がある．というのは，我々は CS を膨大な時間を掛けて学ぶ中でこれらを身に付けてきた．しかし，初中等教育の中で「情報」にかけられる時間は限られており，一方で CS の教えはそのエッセンスを述べただけで身に付くようなものではないと考えられるからである．

課題は山積みだが，まずはこのような課題があることを多くの人に知ってもらうこと，そして現在改訂を進めている[3]「情報教育課程の設計指針」[4] をはじめ，カリキュラム検討のもととなる文書に組み込むことを目指して活動していきたい．

参考文献

1) 日本学術会議：大学教育の分野別質保証のための教育課程編成上の参照基準情報学分野 (2016), https://www.scj.go.jp/ja/info/kohyo/pdf/kohyo-23-h160323-2.pdf
2) Wing, J. M：Computational Thinking, CACM, Vol.49, No.3, pp.33-35 (2006), https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf, 中島秀之 訳：Computational Thinking 計算論的思考，情報処理，Vol.56, No.6, pp.584-587 (2015), https://ipsj.ixsq.nii.ac.jp/records/141823
3) 久野　靖：「情報教育課程の設計指針」の改訂について，高校教科「情報」シンポジウム 2024 秋論文集，pp.19-26 (2024).
4) 萩谷昌己：「情報教育課程の設計指針」解説，情報処理，Vol.62, No.4, pp.e61-e68 (2021).

(2025 年 1 月 27 日受付)

久野　靖（正会員） skuno@acm.org

1984 年東京工業大学理工学研究科情報科学専攻単位取得退学．同大学助手，筑波大学講師，助教授，教授，電気通信大学情報理工学研究科教授を経て，同大学特命教授．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．