

情報教育とプログラミング教育: 私達は当事者である(べき)かも?

久野 靖*

2018.3.5

1 はじめに

- 日本の初等中等情報教育
- 各国の情報教育との比較
- 小学校プログラミング教育
- 情報入試と情報入試研究会
- 思考力・判断力・表現力
- プログラミングどうやって教えるの?

2 「情報教育とプログラミング教育」

- 質問: (小学校～大学初年次までの) 情報教育 / プログラミング教育はあなたにとって関心事? (各々聞きます)
 - 情報教育: YES / NO
 - プログラミング教育: YES / NO
- 質問: 自分がプログラミングに目覚めたのはいつ?
 - 小学校時代 (ないしそれ以前)
 - 中学校
 - 高校
 - 大学
 - まだ目覚めていない
- 質問: もし自分がもっと早くから「例: 小学校」プログラミングに目覚めていたら自分はもっと「書ける」人になっていたと思う? YES / NO
 - もし「YES」なら「日本のために」何かした方がいいわけだね?
 - さらに「プログラミングの才能」ある人がプログラミングに触れないままよその分野に行ってしまうのにも…

3 日本の情報教育の経緯

- 1970 改訂高校学習指導要領: 数学一般「電子計算機と流れ図」応用数学「計算機と数値計算」
- 1980 年代 → PC の普及 → 散発的に (教員個人の試みとして) BASIC、LOGO などによるプログラミング
- 1989 告示指導要領: 小学校から高校までの複数科目で情報教育を行なうと記載 (実際には何も行なわれない)
- 1997 年: 文部科学省調査協力者会議 → 高校に独立教科「情報」新設提唱 (新しい教科ができるというのはなかなか大変なこと)
 - 3 目標「情報活用の実践力」「情報の科学的な理解」「情報社会に参画する態度」(科学、社会を中心とする 2 科目を提唱)
- 1999 告示指導要領: 情報科新設 (2003 年実施) → 「情報 A (実践力中心)」「情報 B (科学中心)」「情報 C (社会中心)」の 3 科目 (各 2 単位) から 1 科目以上選択必修
 - 「情報 A」(易しい科目) を追加、「情報 B」にのみアルゴリズムの話題 (「深入りしないこと」「プログラミングはやらない」)
 - 全国の全高校生が学ぶ → 多数の「情報科免許を持つ」教員が必要 → 現職 (他教科) 教員に 15 日間の講習で免許 → 3 年間で 14,269 人を「促成栽培」(つなぎのはずが永続化)
 - A/B/C は本来「生徒のが興味・関心に応じ選択」のはずだが実際には学校がどれか 1 つ選択 → しかも易しい「A」が 8 割「B」は 1 割未満
 - 教科書も Windows の操作画面多用のものが圧倒的シェア → 多くの高校の情報科の内容は「Word/Excel/PowerPoint」(今も?)
 - 未履修問題の発覚 (世界史が発端だが情報でも多数)(入試にない科目だからと軽視)
- 2008 年告示指導要領 (2013 年実施) → 「社会と情報 (旧情報 C)」「情報の科学 (旧情報 B)」の 2 科目 (各 2 単位) 選択必修に (このほか中学校技術で全員が計測・制御プログラミングを少量学ぶ)

*電気通信大学

- プログラミングが含まれている「科学」の比率は3割程度(先生が教えたくない/教えられない)
- 他教科と掛持ちの教員が多い(情報に軸足がない)
- 47都道府県の多くで情報科教員の採用は「0」よくて「1~3」(2017年度: 大阪12、埼玉8、東京・神奈川県6、あとは3以下、34道府県でゼロ)(採用ありでも複免許必要が多い)
- 情報科新設時に多くの大学が設置した免許課程も撤退相次ぐ
- じゃあ教員はどこから来ているの? 非常勤講師(まだまし)、免許外担任許可・臨時免許(免許を持ってない教員に担当させる): 情報が突出して多い

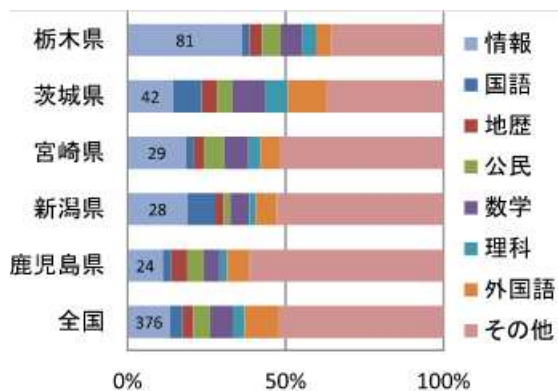


図1 臨時免許状の教科の割合とうち情報科の実数(情報科の臨時免許状の授与件数が多い5県)

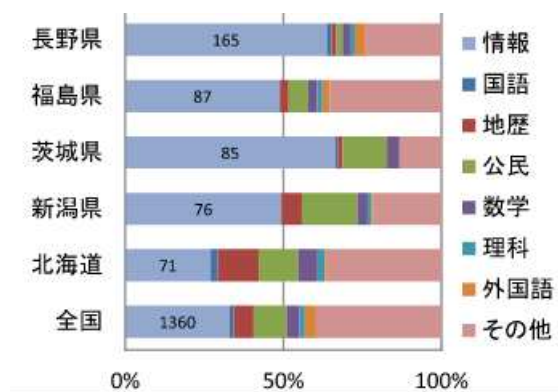


図2 免許外教科担任の教科の割合とうち情報科の実数(情報科の免許外教科担任の許可件数が多い5道県)

(参照) 中山 泰一, 中野 由章, 角田 博保, 久野 靖, 鈴木 貢, 和田 勉, 萩谷 昌己, 箕 捷彦, 高等学校情報科における教科担任の現状, 情報処理学会論文誌教育とコンピュータ (TCE), 3(2), 41-51 (2017-06-14), 2188-4234.

- 2017 告示小学校指導要領: 複数の科目内でプログラミングを扱う(後述)
- 2017 告示中学校指導要領: 技術科で計測・制御に加え「双方向性のあるコンテンツに関するプログラミング」
- 2018 告示高校指導要領: 「情報 I(必修)」「情報 II(選択)」の積み上げに移行(2021-)

- 情報 I「情報社会の問題解決」「コミュニケーションと情報デザイン」「コンピュータとプログラミング」「情報通信ネットワークとデータの利用」
- 情報 II「情報社会の進展と情報技術」「コミュニケーションと情報コンテンツ」「情報とデータサイエンス」「情報システムとプログラミング」+ 課題研究
- 情報 I(必修)は「大学入学共通テスト」に入る予定(2024-)

4 各国の情報教育との比較

□ どちらだと思いますか?

- A: わが国の子供たちは他国よりも情報技術を使いこなしている
- B: わが国の子供たちは情報技術の使いこなさずで他国にたち遅れている

□ OECD PISA 調査の中の ICT 活用調査

- 遠藤 諭さんのサイトから... <http://ascii.jp/elem/000/001/410/1410256/>

□ 「オタクですぎる」(遠藤さんのことば)

- 私から見たら大変うなずける結果

□ 学校で ICT 機器がどういう扱いされてるか知ってます?

- 学校に ICT 機器を持ってきてはいけない
- 学校で用意した ICT 機器しか使ってはいけない
- 先生がいいと言った時だけしか使ってはいけない
- 先生はできる限り使わないようにしたいと思っている

□ 教師暴行動画事件で校長「SNS が悪い」←これも同じこと

□ 他国と比較するといかにひどいか分かる(ex. スウェーデン)

□ 小学校 1 年生からキーボードで入力して物語を作成

- (日本の国語科では手書き絶対。コンピュータとんでもない)

□ 小学校 1 年生から保護者や先生にメール

- (日本の学校では対外連絡可能なメールは与えない)

□ 自分の機器を持って行く=家でも学校でも同じように活用

- (日本ではもちろん持って行くこと禁止)

□ 授業時間中でもいつでもネットで調べてよい

- (日本では「とんでもない!」)

□ 「ICT 教具論」(豊福)

- 日本の学校では ICT 機器は教具（授業をするときの道具）の 1 つ。だから決まった風にしか使ってはいけないし「教材」（特定単元向けの動画とかアプリとか？）が必須
- 他国では ICT 機器は文具（調べたり記録したり整理したり発信したりするのに使うツール）

- 大人になって社会に出た時の使い方はどっち？
- 挽回するチャンス： プログラミング教育
- 英国王立協会のレポート「Shutdown or restart?」（2012）

<https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>

→ 「ICT ツールばかりやってもいいことはない」「プログラミングこそやるべきだ」→世界的なプログラミング教育の潮流

- 英国では 2016 年から小学校で教科 Computing を導入、プログラミング教育を開始
- しかし英国もうまくいっていない

<https://royalsociety.org/topics-policy/projects/computing-education/>

- 高校の選択（GCSE）で学ぶ生徒が少ない
- 性別間格差
- 教員不足/既存教員へのサポート不足
- CS 教育の研究に基づいた政策立案の不足

- すぐに対応： 政府が教員養成に 150 億円投入、Google も 1.5 億円支出
- 日本の場合…高校必修が維持されていて、入試でも採用されていけば希望はあるかと
- 小学校プログラミングは？（次の話題）

5 小学校プログラミング教育

- 各国の動きをうけて文部科学省で有識者会議を招集 → 「小学校段階におけるプログラミング教育の在り方について」（2016.6）
 - ロボット・AI の時代 → コンピュータが魔法の箱でなくプログラムにより動くことを分らせる
 - プログラミング言語は変わる（なくなる?）かも → プログラミング的思考には普遍性がある → これを重視
 - 小学校では「身近な生活でコンピュータが活用されていること、問題の解決には必要な手順があることに気付くこと」が目標

- 2017 告示小学校学習指導要領での記載 → 「プログラミング」は下記の 4 箇所

- 総則「プログラミングを体験しながら、コンピュータに意図した処理を行なわせるために必要な論理的思考力を身に付ける」学習活動を指示（全般的な指示）
- 算数「正多角形の作図を行なう学習に関連して、正確な繰り返し…（中略）さらに一部を変えることでいろいろな正多角形を同様に考えることができる場面などで扱うこと」
- 理科「例えば…電気の性質や働きを利用した道具があることを捉える学習など、与えた条件に応じて動作していることを考察し、更に条件を変えることにより、動作が変化することについて考える場面を取り扱うものとする」
- 総合「プログラミングを体験しながら論理的思考力を身に付けるための学習活動を行なう場合には、プログラミングを体験することが、探究的な学習の過程に適切に位置付くようにすること」

- これを具体的にどうやるのかは不明（というか問題山積み）

- 小学校の先生が自力でプログラミング教育を勉強して実践というのはなかなか大変そう
- 英語教育が圧倒的に注力されていてプログラミングは軽視（周囲の無理解）
- 都会はまだいいが地方では民間とかいってもリソースがない
- 起きていること： 「指導要領にはプログラミングと書いてあるがコーディングは目的じゃないのだから、プログラミング的思考を机上やアンプラグドでやればよいのでは」

- 久野の考え： 「プログラミング的思考を目的とし、プログラミングは目的としない」みたいなのはやめた方が…まずプログラム作って動かすようにならないと分からないのでは？

- 日本の希望が持てる場所は…

- 教育用言語の開発： Viscuit(原田)、SmarlRuby(高尾)、Dolittle(兼宗・久野) など…
- 総務省などもお金を出して事例を作るための事業をやっている（昨年招待講演者の原田さんのところでも多く引き受けている）
- 民間の活用もおこなわれている（小学校の OB がお金を出して等）
- 民間のプログラミングスクール、CodaDojo や Ruby プログラミング少年団などの活動も活発になされている

- 皆様もぜひ機会があればご協力を！

6 情報入試と情報入試研究会

□ 高校における情報科軽視（無視?）の原因： 入試に出ないこと

- センター試験に「情報関係基礎」はあるが多くの大学で利用できない
- 各大学の個別試験でも「情報」はごく少ししか出題されていない
- 利用大学少ない→受験勉強の効率悪い→選ばれない→利用大学少ない
- 慶應 SFC、明治大学情報コミュニケーション学部では最近採用

□ 情報入試研究会 (<http://jnsg.jp>): 2012.1 に有志で立ち上げた団体 (2013 から情報処理学会情報入試 WG、2016 から同情報入試委員会と表裏一体)

- 目的: 情報科の適切な評価を行なう問題が作れることを示すこと、またそのような入試の可能性について広報
- 問題セットを実際に作成し、4 回の模擬試験を実施した
- 会場受験者は少数だったが高校での団体受験は結構多かった

□ 作題では知識問題を避け易いものから難しいものまで「考える」問題を工夫。問題構成は AB の 2 セット (各 45 分: 高校の授業で使える)、それぞれ小問群 (共通)+大問 (科学・社会)

□ set-003(第 2 回) の場合: A セット

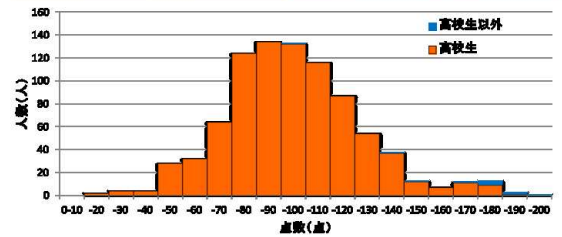
- 大問 1(小問群): 2 進法計算問題、音のデータ量、Web サイトの作成指針、ネットワークサービスの機能
- 大問 2(科学): プログラミングの問題 (短冊型…後述)
- 大問 3(社会): スマホに関するアンケートのデータの解釈

□ set-003(第 2 回) の場合: B セット

- 大問 1(小問群): キーボードのポジション、記録メディア、ファイル形式、GPS の原理、相対 URL の解釈、組み合わせの数
- 大問 2(科学): 寿司店の注文システムのデータ設計と画面遷移
- 大問 3(社会): コミュニケーションを題材とした長文の読解問題

第2回大学情報入試全国模擬試験 結果

全体	高校生	
受験者数	869人	
平均点	95.8点	
標準偏差	27.9点	
	受験者数	858人
	平均点	94.9点
	標準偏差	26.7点



□ 一定の認知は得たが毎年ボランティアでやるのは相当大変

□ 短冊型問題 --- プログラムを書かせる問題を出題するため開発

- プログラミングの試験は (紙でも CBT でも) 「穴埋め問題」が多い ← 採点しやすい
- 穴が正しく埋められるからといってプログラミング能力があることには全くなならない (ex. 基本情報技術者の対策本など)

□ 「紙にプログラムを全部書かせる」のが最善だと思うが…採点がものすごく大変

□ CBT であればプログラムを打ち込ませられるが採点は?

- プロコンのような自動判定 → 「正誤」しか出てこない、cheat 対策必要 (「1+2+…+10 を計算せよ」 「puts(55)」)
- コードを制御フロー/データフロー解析して… → 大変
- 機械学習でよいコード悪いコードの特徴を検出 → 試験採点にはまだ…

□ アイデア: (はるか) 昔、パンチカードにプログラムを打っていた頃、「床にぶちまけたカード」を正しく復元できる人は「書ける人」

- 正解プログラムを行単位で分けて惑わし選択肢を追加し、並べて選択記号を振る → 正しいプログラムを構成し記号列を解答
- DEMO (set-003A 大問 2)

□ 情報入試研究会の各模擬試験で多様な短冊問題を試行

- 複数の (方針の異なる) 正解を許容 → プログラミングの性質と一致
- 採点は自動化容易 (いずれかの正解との編集距離の最小が 0 なら満点、1 なら部分点、など)
- 「プログラミングができる/できない」は見分けられるように思える

7 思考力・判断力・表現力

- ご存じと思いますが： 高大接続改革で大学入試は「知識・技能」に加えて「思考力・判断力・表現力」と「主体性・多様性・協働性」を評価することが求められている
- 文部科学省「大学入学者選抜改革推進委託事業」(DNSKSIJ) 選定「情報学的アプローチによる『情報科』大学入学者選抜における評価手法の研究開発」(代表：大阪大学、連携： 東京大学・情報処理学会)

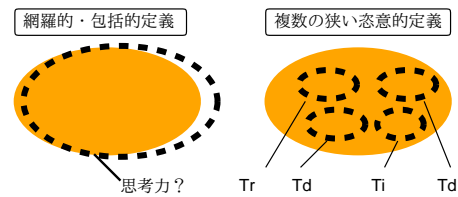
- 情報科での「思考力・判断力・表現力」評価手法の検討を含む←久野担当
- (そのほか CBT 開発とか色々やっています)
- (情報教育の参照基準も…←ポスター発表予定)

- 「思考力」って何ですか???
- 考える力…まんまやんけ!
- 普通に問題作ったら「考える力」を見たことになるの?
- 知識問題(暗記していれば答えられる)は思考力ではないような…
- {すべての問題の集合} - {知識問題の集合} = {考える力の問題}??
- それでは「問題の作り方の指針」にできない orz

- 重要な前提： 「思考力」「判断力」「表現力」の包括的・網羅的定義は行なわない!
- そんなものが出せるなら苦労はしない(大仕事すぎる)
- 「思考」がそんな簡単に定義できるくらいなら自分をもっと偉くなってる(と思う)

- ではなくて…
- T(Thinking/思考力)、J(Judgement/判断力)、E(Expression/表現力)の「恣意的な」「狭い」定義を天下りで勝手に定める
- ただし、たとえば思考力 T なら、ある受験者が T を持つなら、世の中の全般的な理解として、その受験者がその特定面についていえば「思考力」を持つと異論ないだろう、というふうに T を定める
- たとえば： 詰将棋が解ける人は世の中の的にいって「思考力がある」
- なおかつ、T を「問題を作ることが比較的容易であるように」定める
- その T に基づき作題する
- 当然想定される反論： 「その T では測れない思考力 T' がある」

- それに対する回答： 「では、その T' の定義と作問も追加しましょう」
- ただしそれはそれなりに手間なので、作題が可能な範囲でできるだけ汎用的に(広く)最初の T を定めておく方が望ましい



7.1 (Tr) reading

- (自分にとって必ずしも馴染みのない) 記述を読んで意味を理解する力
- 作問方法： 記法の定義やその定義を参照する記述の読解ができていないことを見る問題。
- 作問例： アルファベット A~Z と演算 \diamond および \triangle が混ざって並んだ列を考える。列 s に対し、 $s \diamond$ は s を 2 回繰り返すこと、 $s \triangle$ は s を左右反転することを意味する。 s は空でもよい。演算は左から解釈する。

例： $AB \diamond \triangle \rightarrow ABAB \triangle \rightarrow BABA$

以下の選択肢のうち互いに同じ結果となるものをすべて挙げよ。

- ア BABA \diamond
- イ \diamond A \diamond \diamond
- ウ ABBA \triangle
- エ AB \triangle \diamond
- オ AAAAA \triangle
- カ B \triangle A \triangle \diamond

7.2 (Tc) connection

- (一見関連が分からないところから) 結び付きを見出す力
- 作問方法： 多数の事項の中から結び付きを発見できるか見る設問
- 作問例： 次の文を読み、正人の動作とその理由の組になるものを挙げよ。解答欄は余るかもしれない。

- 「正人は起きて、寒くはなかったが、シャツを着た。春子が来る予定だったので。次に空腹だと思い、パンを食べた。ジャムは塗らなかった。嫌いだったので。」

解答欄： 動作：_____ 理由：_____ 動作：_____ 理由：_____

動作：_____ 理由：_____ 動作：_____ 理由：_____

7.3 (Td) discovery

- (Tcで結び付きを発見したものを含めた事項の集まりに関して) 直接に示されていない事柄を発見する力。事柄としては、次のものが考えられる。
 - 事項どうしの関連が持つ規則・規則性やトレードオフ。
 - 事項に内在する問題・法則・原理。これらは「問題発見」「仮説構築」に相当する。
 - 事項の特性や振舞いを説明する上で有用なモデル化や抽象化。
 - 事項に対する現に記述されているのとは異なる視点。
 - 事項が記述されている範囲(文書等)外のものとの事項との関連。
 - 事項の記述・表現に内在する意図。

□ 作問方法: 事項の記述を与えた上で、上記のような新たな事柄を発見できるかを見る設問

□ 作題例: 次の整数が共通に持つ性質について 20 文字以内で述べよ

5, 9, 12, 20, 33

7.4 (Ti) inference

□ (Tcで結び付きを発見したものやTdで発見したものを含めた) 事項・事柄の集まりに対し推論を適用する力

□ 作問方法: 推論の正しさ判別を見たり、推論そのものを構築させる

□ 作題例: 次の等式すべてが成り立つことはあり得ない。矛盾を生じる最小の集合をすべて列挙せよ

(1) $x = y + 3$ (2) $y = t - 5$ (3) $x = z + 5$
(4) $t = x - 2$ (5) $z = y + x$ (6) $y = z - 4$

7.5 (Ju) judgement

□ (優先順位づけを含め) 複数の事項(トレードオフを含む)の中から、規定した基準において上位ないし下位のものを選択する力。基準としては、次のものが考えられる

- 個数、効率、金額などの理工学的に合理的な指標。
- 社会的、倫理的、道徳的な影響や重要度。
- 制約条件を与えることで順位が変化するような指標(セキュリティ、安全などエンジニアリングデザイン的な指標)

□ 作問方法: 設問によって与えられた事項や、Tcの結び付きの中から、Tdで発見した事柄の中から、あるいはTiの推論の道筋の中から、正しいものや重要なものを選ぶ設問。必要に応じて前提とする状況や制約を付記する。

7.6 (Ju) judgement

□ 作題例: ある職場では顧客から預ったデータが漏洩していたことが発見されるという事件があった。この事態を受けて原因や対処方法を検討するミーティングを開催することになった。そこで使用するのに最も適していると思われる図式を選べ。

(1) フローチャート (2) 状態遷移図 (3) 魚の骨図
(4) PART図 (5) 散布図 (6) ER図

7.7 (Ex) expression

□ (与えられた基準において有用な) 表現を構築/考案/創出する力。基準としては、次のものが考えられる。

- 日本語記述としての適切性(内容が過不足ない、把握しやすい提示順序、適切な接続関係の採用など)。
- 図や絵(グラフや状態遷移図その他特定の図法によるもの、および一般的な模式図や絵の形のもの)・表などで事項を表現する場合の適切性。重要な事項が読み取りやすく表現されているか、アピールするなど。
- 自分や他者の問題解決に資する表現としての適切性(提示された問題の本質的な部分の選択や解決に至りやすい構造の選択など)。
- プログラムなど処理手順記述としての適切性(求める結果の出力や構文規則への合致など)。
- 自分と必ずしも前提が共通しない他者に理解可能な表現としての適切性(コミュニケーション内容としての適切性)。
- SNSやネットなどの場における行動の適切さ(誤解を生まない、他者に迷惑を掛けない、自分や他者にとって価値がある等)。

□ 作問方法: 設問によって与えられた事項や、Tcの結び付きについて、Tdの発見した事柄について、あるいはTiの推論の道筋について、適切な表現を構築する設問。Trの記法や定義(所与のものまたは自分で定める)を適切に活用した記述も含む。必要に応じて前提とする状況や制約を付記する。

□ 作題例: 自転車の利用を促進することが社会的によいという主張をする文章を作成したい。書き出しは「自転車の利用を促進することは社会的に好ましい。なぜならば、」である。これに続く文章の断片を選択肢から選んで記入せよ。句読点は適宜補われるものとする。同じ選択肢を複数回使ってもよい。解答欄は余ってもよいが、解答欄を超えて記入することはできない。

ア 自転車にはタイヤが2つついている
イ 自転車に乗るには技能が必要である
ウ 自転車は人力によって動く
エ 自転車に乗る能力は多くの人か子供時代に身につける
オ 大人になってから自転車に乗ることを学ぶのは大変である
カ 自転車の駐輪が社会問題となっている

- 演習問題を1つでなく多数(易しいものから難しいものまで)提供
 - どれをやるかは学習者が選ぶ(できる人は易しいのは速攻で終わるのでどんどん進むが自分のレベルに相応した問題まで来るとそこで時間が掛かるようになりそれに取り組む)
 - 提出する課題も学習者が選ぶこととする(現在は「1問以上」としている)(できる人が易しい問題だけで済ませないように説得する)
 - 課題は普通のをを各回1問以上出せば合計50点。あと50点が試験。
 - 試験もすべて「プログラムを作らせる問題」と告知(短冊問題)。
 - 試験は短冊問題を多数用意し配点を同一とする(1問2点。できる人が20問40点、苦手な人が10問20点は妥当)(満点は素点の満点でなくできる人の点数)
- できる人もできない人もそれぞれ自分の水準で学んでもらえれば評価するという方針。
- 問題は途中で挫折する人。挫折させないための手当てはまだ必要。
- 電通大の情報基礎科目 --- 2科目(全学必修、700名、13クラス)
- コンピュータリテラシ(前期) --- Unix、ファイル、LaTeX、HTMLなどの内容
- 「コンピュータの原理」でアセンブリ言語(1回)、「ソフトウェアの開発」でJavaScript(1回)を扱う → 後期に備えて準備
- 基礎プログラミングおよび演習(後期) --- 上でのべた方式で実施
- 最初の10回がRuby、あとの5回がC言語
 - 2017年度の経験 --- Rubyは順調だったがCに来たところでギャップに苦しむ
 - C言語に来たところで「離陸し直し」が必要であるというのが教訓
- 評価してみたらどうだったか?
- 評価方式: 短冊(split-paper)テスト ← 自動採点容易だがプログラミング能力が測れると考えている

問題セットの概要と素点の分布: (2コブを免れた!!)

カテゴリ	問題群 A	問題群 B
01. 分岐のないプログラム	引数 x を受け取り、作業変数 y を併用して $x^2 \pm C$ 程度の計算を行なう	引数 x を受け取り、作業変数 y を併用して $x^n \pm x^m$ 程度の計算を行なう
02. 分岐のあるプログラム	引数 x, y, z を受け取り、うち2者の大小に応じ別の値を返す	引数 x, y, z を受け取り、3者の大小関係に応じ別の値を返す
03. 配列を扱うプログラム	配列 a を受け取り、合計や最大などの計算結果を返す	配列 a を受け取り、全部の値が同じか否か等はやや込み入った結果を返す
04. 単純な再帰	再帰を用いて $1+2+\dots+N$ のような単純な結果を求める	再帰を用いて左記と同程度の単純な結果を求める
05. 2次元配列を作る	ほとんどが $1(0)$ で一部が $0(1)$ である 5×5 配列を生成する	多くの要素が 0 で残りに連番が入るような 5×5 配列を生成する
06. 画像の生成	円や長方形を塗るルーチンと呼び重なりのない配置の図を再現する	円や長方形を塗るルーチンと呼び重なりのある配置の図を再現する
07. 整列アルゴリズム	配列の前半と後半を別の配列の前半/後半に正順/逆順でコピー	整列アルゴリズムに現れるような様々な配列の操作
08. モンテカルロ法	擬似乱数を用いて N 回値を生成し簡単な統計値やシミュレーション結果を得る	擬似乱数を用いてやや込み入った規則のゲーム等のシミュレーション結果を得る
09. クラス定義	次々に指定された規則で値を取得できるようなクラスを作る	値を複数投入し、その最大値など投入値に応じた値が取得できるクラスを作る
10. 単連結リストと再帰	単連結リストの指定位置への挿入や削除などを行なう	再帰や相互再帰により数値の単連結リストから奇数番の合計などの値を得る
11. C言語の基本	文字「.」と「*」を連ねて長方形や台形などを出力する	文字「.」と「*」を連ねて左記と同様の形を出力する
12. C言語による計算	三角関数や数学関数の簡単な数表を出力する	指定された数列(例: $a_i = \frac{1}{i}$ など)の和を出力する
13. C言語の文字列操作	文字列 s と文字 c を受け取り、 s に関して c を用いた条件の正否を返す	文字列 s を受け取りさまざまな変形をおこなう(複数の手続き使用)
14. C言語の構造体	構造体を用いた数値対の単連結リストを受け取り指定の結果を求める	2つの実数フィールドを持つ構造体に指定の値を入れて返す

