



Title	学習者中心の教育評価とプログラミング学習 : 学習観モデルの作成と測定をとおして
Author(s)	小野, 哲雄
Citation	年報いわみざわ : 初等教育・教師教育研究, 12: 13-21
Issue Date	1991-03
URL	http://s-ir.sap.hokkyodai.ac.jp/dspace/handle/123456789/8558
Rights	本文ファイルはNIIから提供されたものである。

学習者中心の教育評価とプログラミング学習

— 学習観モデルの作成と測定をとおして —

小野 哲雄

1 はじめに

日本の、学校を中心とした教育システムは、明治以来さまざまな問題をかかえながらもかなりの成果をおさめてきた。しかし、その成果が大きかった分、そのシステムへの過度の依存と固着を生みだす結果となっている。一元的な価値観による偏差値信奉—学歴主義は、社会の硬直化を生み、いま教育の本当の意味が問われている。

その現在、コンピュータが産業の構造を変え、社会をも変革する大きな力を生み出している。それは、当然のように教育のあり方にも影響を与えつつある。機械の発明も人間の生活を大きく変えたが、それは人間の“力”を増大させるものだった。しかし、コンピュータは、人間の力もつ能力であるシンボルの操作に深く関わり、その能力を向上させたものと考えられることができる。したがって、人間の考えること、学ぶこと、教えること、つまり教育に与える影響は産業革命の比ではないのである。

教育システムの形骸化、空洞化が言われるいま、コンピュータは教育に新しい視点をもたらす可能性がある。それは、単にコンピュータを授業に持ち込むことではなく、教育の本質に焦点をあて、見失われていた教育の本当の意味が見えてくる可能性である。

そこで本稿では、現行の教育評価に対する批判をとおして、コンピュータ教育の可能性と新しい教育への方向性を模索してみたい。

2 教育評価

2.1 現在の教育評価の問題点

教育評価は、教育というシステムの中で不可欠の構成要素と考えられている。すなわち、教

育は教育目標を達成するという方向性を持った営みである以上、その成果を確認し、指導に役立てていく必要がある。それを行なうのが教育評価ということになる。しかし、現在の教育評価には、いくつかの問題点があり、それは以下の3点に要約できる。

- a. 評価の対象が、測定しやすい記憶に関わる部分に偏重している。
- b. 総括的評価が中心となっており、教育活動の改善に役立っていない。
- c. 社会における学歴主義から、評価結果だけが重視され、偏差値至上主義を生みだしている。

教育評価とは、本来何のために行うべきものなのであろうか。この本質を捉えた定義として「教育評価とは、教育活動にかかわる意志決定の資料として、教育活動に参加する諸部分の状態、機能、所産などに関する情報を収集し、整理し、提供する過程である」（東、1979）というものがある。つまり、評価とは、広く教育のすべて—学習者、指導法、教材、教師—に対して決定をくだすための情報の収集と利用ということである。現在の教育評価は、この教育改善のために教育情報をフィードバックするという機能に欠けているといえる。

2.2 学習者中心の教育評価へ

以上のように、これまでの教育評価には問題点が多く存在している。このような教育評価の枠を押しつけられてきた生徒たちは、教えられたことをそのまま記憶し、テストでは正誤の結果だけに関心を持ち、生活においてもなるべく失敗しないよう行動する傾向が見られる。また、社会というマクロな視点から見たとき、学歴主義、結果主義、受動的態度は国民の活力や成長をそぐ方向に作用していくのである。

一方、現代においては、情報化社会で生活し

ていくために情報を自分で取捨選択し、加工していく能力の向上、生涯学習の必要性がいわれれており、また個性化による思考の多様性を重視する傾向にある。このことは、学習者が自分で学んでいくという自己学習能力の重要性が増加しているということである。つまり、今後望まれる教育評価は、学習者が自らの学習状況をチェックし、次の学習に役立てていくという学習者中心の教育評価ではないだろうか。

3 学習観

3. 1 学習観とは

このように自己学習能力が強調される今日の教育においては、学習者自身の「学習観」は極めて重要である。学習観とは、「学習とはどのようなものか」に対する学習者自身の認識であり、学習行動において大きな影響をもつ要因である(堀野、1990)。これまでの学校教育において支配的だった伝統的学習観は、「有能でなく、受動的な学び手に専門家が教え、時には学習の到達度をテストによってチェックする」(稲垣・波多野、1989)というものであり、行動主義的教授学習理論により“失敗は悪いことである”とされてきた。しかし、これまで述べたように、今日の教育においては自己学習能力の重要性が増加している。つまり、このための望ましい学習観とは、「学習者が能動的であり、好奇心を持ち、失敗を生かして自ら学んでいく存在である」(稲垣・波多野、1989)となり、認知的な教授学習理論もこの学習観の立場にある。

3. 2 学習観のモデルの作成

以上のような望ましい学習観を形成していくには、どのような要因を必要とするだろうか。そのモデルを作るため、4つの要因を設定する。

- a. 学習に対する自発性・能動性
- b. 失敗に対する柔軟的態度
- c. 思考過程の重視
- d. 自己の能力に対する信頼感

aは、知識をただ教えられるという受動的な態度ではなく、自分から興味を持って取り組んでいくという自己学習能力の基本となる要因である。「学ぶこととは、学習者がその必然性を把握し、主体性を持って活動することが基礎と

なる」(三宅、1985)のである。bは、教育において重要なことは失敗させないことではなく、失敗に対して柔軟な態度を育成し、そこから多くのことを学んでいこうという考えである。cは、学習の結果のみに執着することなく、その結果に至るまでの過程について自分で思考することが重要であり、また楽しいものであるという姿勢を身に付けることである。この2つの要因の重要性については、パパート(1982)がプログラミング言語LOGOを使った教育によって主張しており、さらに堀野(1990)は、この要因に実験的考察を加えている。dは、自分の能力を信じ、「やればできる」という考えを持つことである。自分の能力を生まれつき変化しないものと考え、努力に対して消極的な態度を生じさせ、無力感を持たせることになる(水野、1987)。ゆえに自己の能力に対する信頼感を持つことが、学習者を新しい学習へと動機づけていくことになるのである。

この4つの要因は、当然重複した部分を含んでいるが、今後必要と考える自己学習能力を身に付ける上で、それぞれ必要かつ独自の要因であると考えられる。

4 学習観を育てるプログラミング学習

以上、学習者中心の教育評価から望ましい学習観のモデルへと議論を進めてきた。それでは、その学習観を育てるにはどうしたらよいのであろうか。ここでは、コンピュータ利用教育が学習観を育てる可能性について考えてみたい。

4. 1 コンピュータ利用教育

コンピュータを利用した教育には、3つの側面が考えられる。

- a. コンピュータリテラシーの教育
(teaching about computer)
- b. 学習や自己表現の道具としてのコンピュータの利用
(teaching with computer)
- c. 教授のための道具としてのコンピュータの利用
(teaching through computer)

aは、コンピュータそのものを教育の内容として取り上げ、それについて教えるものである。

狭義のコンピュータリテラシーの教育、コンピュータウェアネス教育、コンピュータ科学の教育といわれ、現在でもカリキュラムなどに取り入れ積極的に教育が行なわれている学校もある。bは、コンピュータを生徒の学習や自己表現の道具として使うことである。文書作成、作画、計測、資料検索や実験、シミュレーションをとおして現象のなかに潜む法則性の発見などが考えられている。cは、コンピュータを、教師が教える道具として使うことである。教材の提示、教科教育などに用いられ、いわゆるCAI、CALなどがこれに含まれる。

4. 2 プログラミング学習の可能性

ここでは、コンピュータを道具として利用すること、つまりプログラミング学習を道具として用いることにより、生徒が望ましい学習観を育てるという可能性について考えてみたい。

プログラミング学習の特徴をまとめるとつぎのようになる。プログラミングは決まりきったひとつの正答というものはなく、自由に考え、試してみることができる。また、コンピュータは、自分からはたらきかけなければ何の応答も得られず、はたらきかけによって即時に応答が得られる。このことから、考えること自体が楽しいという「自律的思考傾向」を促進する可能性を持つのではないか（市川、1988）という議論がある。

それでは、プログラミング学習が、さきに述べた“望ましい学習観”を育成する可能性についてはどうであろうか。プログラミングという作業は、自分からはたらきかけなければコンピュータから何の応答もなく、すべてのはたらきかけには即時に結果がフィードバックされる。このことが「学習に対する自発性・能動性」を養うと思われる。さらに、試行錯誤しながらプログラムを作成していく過程から「失敗に対する柔軟的態度」が養われ、また、さまざまな正答が存在することから、結果よりも「思考過程を重視」する傾向を持つようになると思われる。最後に、教師の適切な支援のもとで成功感を得ることにより「自己の能力に対する信頼感」を育てることが可能となってくると思われる。

従来、学校教育で教えてきたものは、事実をあたえそれを記憶させることに重点が置かれた

宣言的知識であった。しかし、現代においては、過去の知識をそのまま適用できる分野は減りつつあり、その状況により臨機応変な対応が必要とされている。また、「人の教育は、彼が、学校で習った勉強を忘れてしまったところで終わる」（永野、1987）といわれ、学校で習った細かいことを全部忘れた後に残るものが本当の教育の成果であるとする立場もある。つまり、学校での“訓練”によって身に付けたことは、社会に出たころにはほとんど役に立たないということである。この意味からも、プログラミング学習は物事を分析・整理し処理していく過程、つまり手続的知識の習得を目指すものである以上、さきの学習観の形成と共に、今後の教育への指針をあたえるものであると考えられる。

5 道具的プログラミング学習

プログラミング学習を教育のための道具として用い、学習観を育てる可能性について述べてきたが、ここではその具体的な学習方法について考えてみたい。

人間の学習は、既有知識を使って対象を理解していく過程と考えることができる。つまり、現在自分の持っている知識を「道具」として学習対象を理解していくことである。ピアジェのいう同化とは、この既有知識を用いて外部のものを取り入れるはたらきであり、調節とは、この既有知識ではうまくいかないとき、この知識を修正して外部のものを取り入れることである。

このような視点から、学習観を育てる道具としてプログラミング学習を考えた時、それは3つのステップに分けることができる。

〈ステップ1〉プログラミング言語の習得

道具として使用するにあたって、プログラミング言語を習得しなければならない。しかし、プログラミング言語のように人工的に形式化された言語は、現実世界との接点がないため具体的なイメージとして捉えられない。つまり、既有知識を利用することが難しいため、学習を困難と感じる学習者が存在する。

〈ステップ2〉学習観の育成

プログラミング言語により、さまざまな課題をプログラミングし、問題解決していく。しか

し、どのような課題でもよいということではない。その課題を解こうとすることが、学習者にとって意味のある環境、つまり「機能的学習環境」(三宅、1985)である必要がある。さらに、課題には、プログラミングの過程から一般化できる概念(再帰処理、構造化など)を含んでいなくてはならない。このような学習をとおして学習観を育てるのであるが、学習者はこの対象を意識してはいない。つまり、学習者は、プログラミングという学習に集中することにより学習観を養っているのである。

〈ステップ3〉学習観から般化

獲得された学習観は、その後の学習、思考そして行動に対して大きな影響を与えていくと考えられる。この時点で、自己学習能力もかなり育成されている可能性がある。

ステップ1については、小野(1990)がプログラミング言語をひとつのスクリプトで読み解くことによりイメージ化を可能にし、その習得を容易にする学習法を実践している。そこで本研究では、ステップ2を中心にこれらの仮説の検証を試みたい。

6 調査と実験授業

本調査と実験授業は、札幌ソフトウェア専門学校応用情報工学科1・2年生68名(男子62名、女子6名)を対象に1990年12月に行なった。この学生は、専門学校入学後プログラミングの学習をしており、1年生はCOBOLを、2年生はCOBOL、FORTRAN、アセンブラ、Cなどの言語を使用してきた。

6.1 学習観に関する質問紙調査

学習観を先に定義した4つの要因から形成されると考え、それぞれの概念を含んだ質問項目を用意し、学習観の尺度構成の測定を試みた。なお、質問項目はそれぞれ7項目とし、そのうち3項目は反転項目とした。これらの項目に対して、「よくあてはまる(5点)」から「まったくあてはまらない(1点)」までの5件法で評定を求めた。対象は上記1・2年生68名であった。

6.2 プログラミングに関する実験授業

実習の時間を使い、次の3題の課題の中から

1題選択し、プログラミングを行ない、プログラムリストを提出するように指示した。この課題自体は、機能的な学習環境の中にあるとはいいがたいが、専門学校という職業教育の場では、学習者にとって意味をもっていると考えられる。なお、課題は難易度の低いものから高いものの順になっており、どれを提出してもよいとの指示を与えた。また、使用言語はC言語とし、上記2年生29名を対象に実施した。

〈課題内容〉(付表1参照)

- 課題1 実数型1次元配列のデータの中から最小値を求める関数を作成する。
- 課題2 整数型2次元配列の中からある値 x を探索し、その大きさの順位を求める関数を作成する。
- 課題3 リスト構造のデータをポインタを用いてソート(整列)する関数を作成する。

7 結果

7.1 学習観に関する質問紙調査の結果

学習観に関する質問紙調査の結果は表1～表4のとおりであった。この結果から、各項目の得点にばらつきが見られるため項目分析を行なった。項目分析は、4要因(「学習に対する自発性・能動性(B1)」、「失敗に対する柔軟的態度(B2)」、「思考過程の重視(B3)」、「自己の能力に対する信頼感(B4)」)それぞれについて、ある項目とその項目を除いた他の項目の合計点との相関係数(項目相関係数)を算出した。その結果、B1要因の「勉強は、自分でするよりも誰かに教えてもらいたいと思うことが多い」とB4要因の「テストで問題が解けないのは、自分の能力のせいだと思う」の項目相関係数が有意ではなかった(表1、表4)。このため、この項目を除外した後、それぞれの項目群の平均値を各要因の尺度得点とした。

以上の結果をもとに学年別(1年と2年)の尺度得点をもとめたが、統計検定の結果、両群に有意な差はなかった。このことは、プログラミングの経験年数が学習観尺度得点に影響を与えないという可能性も示唆されるが、この点は後の考察であらためて取り上げたい。

表1 学習に対する自発性・能動性（B1）要因の質問紙調査結果

項目内容	平均値 (X)	標準偏差 (SD)	項目相関係数 (r)
自分から進んで問題を見つけ考えるのが好きだ。	2.70	1.03	.63*
(-) 難しいことを考えるのは、自分には向いていないと思う。	3.02	1.09	.55*
問題が出されると、何としても解答しようとするほうだ。	3.20	.97	.45*
これからもたくさんのことを学びたいと思う。	4.12	.99	.48*
(-) 勉強は、人に言われてからすることが多い。	2.65	1.29	.34*
ひとつのことを学ぶと、さらにたくさんを知りたくなるほうだ。	3.70	1.01	.54*
(-) 勉強は、自分でするよりも誰かに教えてもらいたいと思うことが多い。	2.61	1.23	.23

注1) (-) のついた項目は反転項目であり、この数値は反転後のものである。 *p<.01

注2) 項目相関係数：ここでは、各項目とその項目を除く合計得点の相関係数を表す。

表2 失敗に対する柔軟的態度（B2）要因の質問紙調査結果

項目内容	平均値 (X)	標準偏差 (SD)	項目相関係数 (r)
失敗を繰り返しながら、だんだん完全なものにしていけばよいと思う。	3.97	.96	.62*
(-) うまくいきそうもないと感じると、やる気がすぐなくなってしまう。	2.77	1.07	.48*
失敗しても、いつかうまくいくはずだと思う。	3.86	.97	.43*
(-) 難しい問題ができないと、すぐあきらめてしまうほうだ。	2.74	1.09	.57*
思ったようにいかないときは、その原因をつきとめようとする。	3.17	1.04	.47*
テストでの間違いは、とても自分の勉強になると思う。	3.50	.97	.57*
(-) 失敗すると困るので、難しい問題は解答しないようにしている。	3.80	1.05	.44*

注1) (-) のついた項目は反転項目であり、この数値は反転後のものである。 *p<.01

注2) 項目相関係数：ここでは、各項目とその項目を除く合計得点の相関係数を表す。

表3 思考過程の重視（B3）要因の質問紙調査結果

項目内容	平均値 (X)	標準偏差 (SD)	項目相関係数 (r)
(-) テストでは、途中の考えより、答えがあっていたかが気になる。	2.41	1.22	.27**
まだ考えている途中なのに、人から答えを言われるのはいやだ。	3.91	1.20	.37*
答えがあっているかどうかだけでなく、考え方があっているかが大切だと思う。	3.76	1.17	.56*
ある問題が解けた後でも、別の解き方をさがしてみることがある。	2.77	1.11	.30**
(-) なぜそうなるのかわからなくても、答えがあっていればよいと思う。	3.53	1.28	.66*
テストで、できなかった問題は、後からでも解き方を知りたい。	3.36	1.07	.53*
(-) 答えがあっていれば、考え方が間違っても気にならないほうだ。	3.39	1.18	.59*

注1) (-) のついた項目は反転項目であり、この数値は反転後のものである。 *p<.01 **p<.05

注2) 項目相関係数：ここでは、各項目とその項目を除く合計得点の相関係数を表す。

表4 自己の能力に対する信頼感（B4）要因の質問紙調査結果

項目内容	平均値 (X)	標準偏差 (SD)	項目相関係数 (r)
努力すれば、勉強はできるようになると思う。	4.27	.88	.54*
(-) 能力は生まれつきのものであり、変わることはないと思う。	3.39	1.36	.53*
(-) どんなに努力しても、できないことがたくさんあると思う。	2.55	1.13	.46*
人間の能力は、自分で高めることができると思う。	3.91	.96	.58*
何事もやればできると考えることが多い。	3.80	1.04	.50*
テストで、できなかった問題は、ただ勉強不足が原因であると思う。	4.15	.94	.27**
(-) テストで問題が解けないのは、自分の能力のせいだと思う。	2.71	1.39	.20

注1) (-) のついた項目は反転項目であり、この数値は反転後のものである。 *p<.01 **p<.05

注2) 項目相関係数：ここでは、各項目とその項目を除く合計得点の相関係数を表す。

7. 2 プログラミングに関する実験授業の結果

プログラミングに関する実験授業の結果、各課題の提出状況は、課題3提出(A1)3名、課題2提出(A2)7名、課題1提出(A3)13名、課題提出せず(A4)6名であった。提出されたプログラムリストは、それぞれ正しく作動するものであった。

この結果から、A1~A4の4群の学生について、それぞれ学習観尺度得点の平均をもとめると図1のようになった。この4群の学習観尺

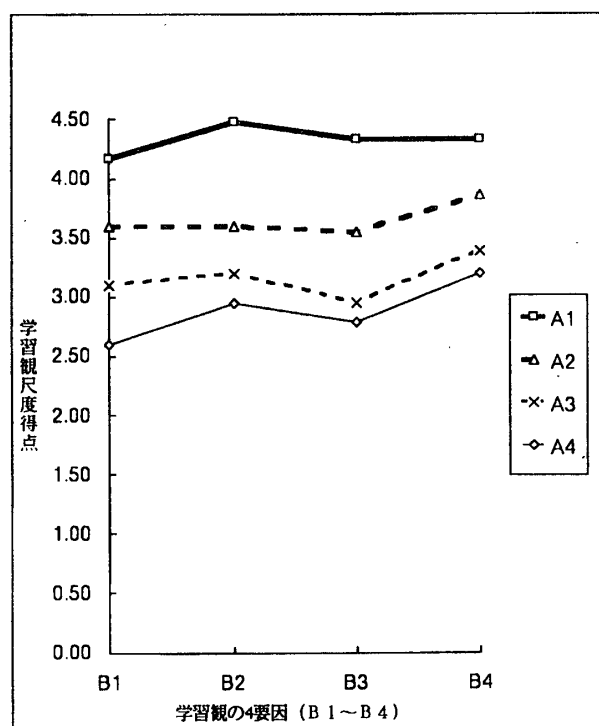


図1 A1~A4群の学習観尺度

度得点を統計分析したところ、A1~A4群の差、および学習観B1~B4要因の差が有意であった(表5)。さらに、A1~A4群間の差の検定をしたところ、A1-A3間($p < .05$)、A1-A4間($p < .01$)に有意な差があった。このことは、学習観尺度得点の高い学生は、より難易度の高い問題に取り組み、なおかつその問題の解答を作成する高いプログラミング能力を持つと考えられる。

また、学習観の4要因については、要因間の相関係数をもとめた。その結果、4要因相互の相関係数は.78~.88の間にあり、それぞれ有意であった($p < .001$)。このことは、学習観4要因の尺度得点間に平均値の差はあるが、その得点は同一傾向を示しているといえる。つまり、この4要因は、学習観という共通の構成概念を測定していると考えられる。

7. 3 学習観尺度得点と高校までの学業成績

プログラミングの実験を行なった2年生を対象に、学習観尺度得点と専門学校入学時の英語・数学のテスト(高校までの履修内容を出題している)の得点との相関をもとめた(表6)。この結果、英語とB1要因の相関が有意であっ

表6 学習観尺度得点と高校までの学業成績の相関

	学習観の4要因			
	B1	B2	B3	B4
数 学	.30	.16	.20	.24
英 語	.41**	.35	.36	.28

** $p < .05$

表5 分散分析表

変 動 因		SS	df	MS	F
A1~A4群の差	A	21.14	3	7.046	3.63**
B1~B4要因の差	B	2.33	3	0.777	4.10**
交互作用	AB	0	9	0	
個人差	C	48.48	25	1.939	
交互作用	BC	14.20	75	0.189	
全 体		86.15			

** $p < .05$

た ($p < .05$) が、その他の学習観尺度得点と英語・数学の得点の相関は有意ではなかった。このことは、本調査で測定を試みた学習観が、高校までの学業成績に影響を与えるものではなかったと考えられる。

8 考察

以上の結果から、本質問紙調査はさきに設定した学習観の4要因をほぼ測定することができたと考えられる。また、実験授業の結果から、プログラミング能力の高い学生は、学習観尺度得点においても高い値を示した。しかし、その学習観尺度得点は、高校までの学業成績とほとんど相関がないことがわかった。このことから、次の二つの仮説が考えられる。

まず第一に、プログラミングを学んだことにより、望ましい学習観を形成したとするものである。高校まで記憶偏重の結果主義、偏差値中心主義の教育により自己の能力を発揮できなかった学生が、プログラミングの学習により失敗を恐れず自分で考えることの楽しさ、やればできるのだという考えを持ち始めたのではないだろうか。

第二は、以前から望ましい学習観を持っていたが、それが学業成績に結びつかなかった、つまり評価されなかったというものである。プログラミングが、その学習観を率直に表現させたとも考えることもできる。今回の調査においても、プログラミング経験の差（専門学校1年生と2年生）による学習観尺度得点の違いは見られなかった。このことは、プログラミング学習が望ましい学習観を育てたというよりも、既に持っていたものを発現させたと解釈することができる。どちらにしても、プログラミングが今までの教科では伸ばすことができなかった能力を育てたという可能性は大きいと考えられる。

しかし、プログラミングの効果を安易に一般化することへの批判もある（三宅、1987）。なぜなら、プログラミングがなんらかの認知能力を伸ばすとするならば、そもそもプログラミングがどういう認知過程なのかわかっていなければならぬからである。その点を無視しての一般化は無理がある。

また、今回の調査、実験授業は、プログラミングのエキスパートを養成する専門学校生を対象とした。このため、一般の学校での教育目標とはかなり異なっているといえる。一般の学校においては、プログラミングの専門家を養成する必要はない。プログラミングを道具として、今まで既存の教科では養い得なかった能力を身につけさせることが目標となってくるであろう。

したがって、一般の学校においては、プログラミング学習を今までのような評価の枠で捉えてはならない。そうすることは、プログラミング学習の良さを奪うことになってしまうからである。基本的に、一般の学校におけるプログラミング学習では、今までのような評価を行なう必要はないと考える。あくまでもコンピュータやプログラムは、学習のための道具であるとし、その学習は一般的な能力の育成に役立っていると考えていくべきである。しかし、学習の改善のため、学習者に教育情報をフィードバックすることは必要である。このことが教育評価の本来の目的である教育活動の改善へとつなげ、学習者に自己学習能力をつけさせるという学習者中心の教育評価へ向かうと思われるからである。

9 おわりに

本稿では、望ましいと考えられる学習観とプログラミング学習との関連について調べてきた。この2つは、相互にかなりのつながりがあると思われるが、明確な結論を出すには今後の研究が必要である。

しかし、「学習観」という視点から教育を見直すと、今まで「知識」や「偏差値」ということばでばやけていた児童、生徒—学習者—の姿がはっきり見えてくるように思われる。そして、これからの教育の方向性も見えてくるようである。

謝辞

本研究を行なうにあたって、ご指導いただいた北海道教育大学助教授（岩見沢分校）鹿内信善先生、およびシステムソフト販売(株)札幌支店長 蝦名信英氏に感謝致します。

（札幌ソフトウェア専門学校主幹講師）

参考文献

- パパート, S. 1982 奥村貴世子訳 「マインドストーム—子ども、コンピュータそして強力なメディア」 未来社
- 三宅なほみ 1987 コンピュータを教える—コンピュータリテラシー— 「岩波講座教育の方法10 教育と機械」 岩波書店 pp120—159
- 三宅なほみ(編) 1985 「教室にマイコンをもちこむ前に」 新曜社
- 永野重史 1987 学ぶことと評価 「岩波講座教育の方法1 学ぶことと教えること」 岩波書店 pp164—202
- 堀野 緑 1990 基本的学習観の測定の試み—失敗に対する柔軟的態度と思考過程の重視— 教育情報研究 VOL. 6 No. 2 pp3—7
- ホルト, J. 1987 大沼安史訳 「教室の戦略」 一光社
- 橋本重治 1979 「新・教育評価法概説」 金子書房
- 梶田叡一 1983 「教育評価」 有斐閣
- 渋谷恵一 1987 「子どもを伸ばす評価」 ぎょうせい
- 小野哲雄 1990 イメージ化スクリプト学習—プログラミング教育への認知心理学からのアプローチ 日本教育情報学会 第6回年会研究発表集録 pp165—166
- 東 洋 1979 「子どもの能力と教育評価」 東京大学出版会
- 市川伸一 1988 コンピュータ教育と精神発達 ころの科学 16 pp14—21
- 稲垣佳世子・波多野誼余夫 1989 「人はいかに学ぶか」 中公新書

付表1 プログラミング実験授業の課題

プログラミング実習課題

下の課題より1題選択し、プログラムを作成しなさい。

注意

- ・問題は、やさしい問題から難しい問題の順に並んでいる。
- ・どの問題を選ぶかは、自分で決めること。
- ・プログラムは、作成途中であってもいいから時間内に提出すること。

課題1

次のような実数型配列 data [] から、最小値をもとめて返す関数 Search (data) を作り、以下のデータを使い実行しなさい。

3.0	5.51	2.0	45.3	17.5	4.82
-----	------	-----	------	------	------

課題2

次のような整数型の2次元配列 data[] からデータ x を探索し、x が存在すればその順位（小さいほうから何番目か）を返し、存在しなければ（-1）を返す関数 Search (data, x) を作り、以下のデータで実行しなさい。

2	8	3
7	5	1
4	9	6

課題3

次のようなリスト構造のデータを、ポインタだけを変更しソートする関数 Sort (data) を作り、下のよう出力しなさい。（リストは、常に0番地から始まり、終わりはポインタが99である。）

データの番地	内 容 infor	次の番地 pointer
0	10	4
1	30	2
2	50	99
3	20	1
4	60	5
5	40	3

→

0	10	3
1	30	5
2	50	4
3	20	1
4	60	99
5	40	2