

C++の学習の問題点に関するアンケート調査 — 技術者と学生を対象として —

菅原 千穂子 †, 植田 佐知子 †, 高田 義広 †, 久保田 益史 ††, 鳥居 宏次 †

†奈良先端科学技術大学院大学 情報科学研究科

††オムロン株式会社

〒 630-01 奈良県 生駒市 高山町 8916 番地の 5
chihok-s@is.aist-nara.ac.jp

〒 525 滋賀県 草津市 西草津 2-2-1
kubota@eftses.krc.omron.co.jp

あらまし オブジェクト指向プログラミング言語 C++の学習を支援することを目的として、現状の学習の問題点をアンケートにより調査した。調査対象は、C++の演習コースを終了した直後の技術者と学生の合計 31 人である。調査項目は、習得の困難な学習項目、学習を支援すべき項目、その支援の方法などについてである。主な結果として、(1) friend, 参照などの学習項目が困難であると認識されていること、(2) それらを含むいくつかの項目が学習を支援すべき項目の候補であること、(3) サンプルプログラムを読むことの効果が高く認識されていることなどがわかった。

キーワード オブジェクト指向, プログラミング言語, 学習, ソフトウェア教育, C++, アンケート調査

A Questionnaire Survey on Problems of Learning C++ — On Both of Software Engineers and Students —

Chihoko Sugahara†, Sachiko Ueda†, Yoshihiro Takada†, Masunori Kubota††, and Koji Torii†

†Graduate School of Information Science
Nara Institute of Science and Technology

††Omron Corporation

8916-5 Takayama-Cho, Ikoma-Shi, Nara 630-01
chihok-s@is.aist-nara.ac.jp

2-2-1 Nishikusatsu, Kusatsu-Shi, Shiga 525
kubota@eftses.krc.omron.co.jp

Abstract We conducted a questionnaire survey on current problems of learning C++, a object-oriented programming language, for the purpose of aiding learners. The survey was conducted on engineers and students, in total, 31 people, immediately after they had finished a training course of C++. The scope of the survey covers various questions such as what items are difficult, learning of what items should be aided, and how we should aid learners. As a result, we found that (1) learners think items such as 'friend' and 'reference' difficult to learn, (2) including the items, several items have priority with respect to learning aid (3) learners think reading sample programs highly effective in learning.

key words Object-orientation, Programming language, Learning, Software education, C++, Questionnaire

1 はじめに

オブジェクト指向によるソフトウェア開発技術の実用化が近年、急速に進んでおり、OS、GUI、ネットワークなどの計算機システムの基本的な機能を担う部分が、オブジェクト指向ソフトウェアとして実現されることが多くなっている [5]。そのため、このような基本的な部分やそのアプリケーションを開発するソフトウェア技術者に対して、オブジェクト指向プログラミングを行うことの要求が高まっている。そして、ソフトウェア開発組織や教育機関において、オブジェクト指向プログラミングの訓練や教育の要求が急速に高まっている [2]。

ところが、オブジェクト指向のプログラミングや言語は、その有用性が認められている一方で、従来のプログラミングや言語に比べて、習得が困難であるという意見を聞く。我々は、大学院などにおいて、プログラミングや言語の教育に携わってきたが、Cにオブジェクト指向の概念を導入したC++[1]について、そのような意見を多く聞いており、その困難さを強く感じている。オブジェクト指向技術の訓練における同様の問題は、文献 [4] でも述べられている。今後、オブジェクト指向のプログラミングや言語の訓練や教育は、重要な問題になると思われる。

以上の問題に対して、我々は、C++のプログラミングについて、学習を効率的に支援することを考えており、そのために、次の3つの主題に関するアンケート調査を行った。本稿では、その結果を報告する。

主題 1 C++の学習項目の中でどれが困難であるか?

主題 2 学習項目の中のどの学習を支援するべきか?

主題 3 どのような方法で支援するべきか?

2 調査方法

2.1 対象

調査は、2つのグループの合計31人に対して行った。どちらのグループについても、C++の演習コースを受けてもらい、コースの終了直後に、同じアンケートに回答してもらった。1つ目のグループは、計算機組み込みシステムを主に開発している企業のソフトウェア技術者17人である。2つ目のグループは、奈良先端大 情報科学研究科 博士前期課程の学生14人である。

演習コースは、グループ別に独立に実施された。理想的には、両グループの全員に同一のコースを受けてもらう方がよいと考えられるが、それは実現できなかった。どちらのコースも、オブジェクト指向の

基本的な概念やC++の講義と、いくつかの小規模な課題に対するプログラミング演習とからなる。オブジェクト指向の概念については、どちらのグループでも、RumbaughのObject Modeling Technique (OMT)の本 [3] に基づく講義が行われた。C++については、特定の本によらず、スライドを使った講義が行われた。演習については、企業の技術者の参加者がそれぞれ5~10回、そして、学生の参加者がそれぞれ2回行った。課題の中には、酒屋在庫管理問題 [7] が含まれており、どちらのグループの参加者も必ずその課題の演習は行った。

2.2 質問項目

アンケートに含めた質問は、次の3種類である。

- 回答者の特徴を知るための質問
- C++の習得の困難さを確認するための質問
- 3つの主題に関する情報を得るための質問

実際のアンケートは、本報告に付録として載せているように、15問からなり、回答に要する時間の平均が20分程度の量であった。以下では、3種類の質問のそれぞれについて詳しく述べる。

2.2.1 回答者の特徴に関する質問

各回答者の特徴や経歴を知るために、使用している計算機環境 (問 1, 6)、普段のプログラミングスタイル (問 2)、演習前に学習していたプログラミング言語 (問 3)、C言語についての演習前の技能 (問 4)、オブジェクト指向やC++についての演習前の知識や経験 (問 5, 7) を質問した。どの質問も、付録に示すように、選択式にした。

2.2.2 C++の習得の困難さに関する質問

C++の習得が困難であるという学習者の主観的な意見を聞くが、それを示すデータは、我々の知る限り報告されていない。そこで、そのようなデータを得るために、習得に必要な日数 N (問 13) と、必要性を考慮した上で習得に費やしてもよいと思う日数 D (問 14) とを質問した。もし、習得に必要な日数が、費やしてもよい日数よりも長ければ、習得が困難であると認識されていると考えられる。逆に、必要な日数が、費やしてもよい日数よりも短ければ、習得がそれほど困難でないと認識されていると考えられる。

2.2.3 各学習項目の困難さに関する質問

主題 1 については、C++の基本的ないくつかの概念 (問 10B) とオブジェクト指向の基本的ないくつかの概念 (問 9B) についての、学習の困難さの順序を質問することを考えた。しかし、学習の困難さ

を直接的に質問しても、各回答者の採った学習の方法の違いなどにより、正しい回答を導き出せない可能性がある。そこで、各項目の学習の困難さは、その概念の直観的な理解や説明の困難さに近いと考えて、その順序を質問するように工夫した。具体的には、表3、表5の項目をそれぞれ列挙して、「説明しにくいと思われる順に並び替えて下さい」と指示した。

2.2.4 学習を支援すべき項目に関する質問

主題2における各項目の支援の必要性は、その項目の習得の困難さと習得後の有用性という2つの特質に影響されると考えられる。習得が困難であっても有用性が低い場合は、必ずしも学習を支援する必要がないと考えられる。有用性が高くても現状の方法で容易に習得できる場合も同様である。そこで、各項目について、学習の困難さに関する質問に加えて、習得後の有用性の順序を質問した(問9C, 問10C)。具体的には、困難さに関する質問と同様に、表6、表8の項目をそれぞれ列挙して、「役立つと思われる順に並び替えて下さい」と指示した。

2.2.5 支援の方法に関する質問

主題3については、どのような方法が効率的であるかを直接的に質問しても、学習に直面していた者から回答を得ることは困難であると思われる。そこで、間接的ではあるが、より具体的な次の3点を質問した。

- 演習コースにおいて問題であると思ったこと(問11)
- 演習コースにおいて行った作業の中で効果的であったこと(問8)
- あればよいと思う支援ツール(問12)

どの質問でも、回答をより容易にするために、いくつかの候補を列挙し(表9、表10、表12)、質問に当てはまる候補から当てはまらない候補へ並び替えるよう指示した。そして、列挙した候補の他のものがあれば、それを書き足して並び替えてもらうよう指示した。

2.3 配布と回収

アンケートは、演習コースの終了直後に、参加者全員に電子メールで配信した。そして、数日以内に、回答を返信してもらった。企業の技術者については、22人に配信し17人から返信があった。学生については、15人に配信し14人から返信があった。回答率は、それぞれ、77%と93%であり、全体では84%である。

3 調査結果

3.1 回答者の特徴

回答者の特徴に関する質問(問1,4,6,7)について、結果を表1に要約する。表では、学生と技術者のそれぞれのグループについて、最も多く選択された選択肢を示している。

表1: 回答者の特徴

	技術者	学生
プログラミングの主な対象	組み込みプロセッサ(4)	UNIX系(2)
C言語のプログラミング	本を見なくても書ける(3)	本を見れば書ける(2)
使用するC++の環境	Microsoft Visual C++(2)	UNIX上の処理系(1)
オブジェクト指向の演習前の経験	学んだことがない(1)	練習で使ったことがある(3)
C++の演習前の経験	学んだことがない(1)	練習で使ったことがある(3)

3.2 C++の習得の困難さ

C++の習得に必要なと思われる日数 N (問13)に対して習得に費やしてもよいと思われる日数 D (問14)が短い傾向のあることがわかった。それぞれの平均を表2に示す。 $N < D$ と回答した者は、31人中の5人しかいなかった。 $N > D$ と回答した者は12人、 $N = D$ と回答した者は14人であった。片側の符合検定[6]を行うと、10%の危険率で $N = D$ が棄却される。学習者の見積もりや希望よりも、C++の習得には時間がかかり難しいと認識されていると言える。

表2: 学習の日数

	技術者	学生	全体
必要な期間(日)	37.4	17.9	28.6
費やしてよいも期間(日)	22.1	14.6	18.7

3.3 各学習項目の困難さ

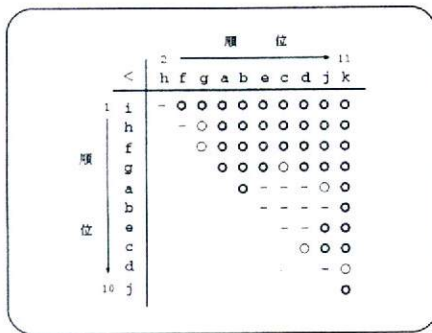
C++の基本的な概念として列挙した学習項目(問10B)については、認識されている困難さによって比較的明確な順序を付けられることがわかった。表3に、その順序を示す。これは、31名の付けた順序にできるだけ矛盾しないように求めた順序である。具体的には、回答の順序において逆順になる項目の対を数えた時に、その総数が最小となるように求めた。なお、演習コース後に全ての回答者が全ての項目を知っていた訳ではない。各回答者には、知らなかった項目を除いて順序を付けてもらい、表の順

序は、そのことを考慮して求めている。表右列の認知度は、各概念を知っていた回答者の割合を示している。各項目の対の間で順位の違いがあるかどうかについては、別に検定を行なった。結果を表4に示す。○印は危険率が10%、そして、◎印は危険率が5%の片側の符号検定[6]によって、順位差に差がないという検定仮説が棄却されたことを表す。表4より、friend、参照、演算子のユーザ定義、関数のオーバーローディングなどは、比較的難しいと認識されていることがわかる。デフォルトの引数については、比較的易しいと認識されていることがわかる。残りのa, b, e, d, jの間では、ほとんどの対に有意な差が見られなかった。

表3: C++の学習項目の困難さの順序

順位	学習項目	認知度 (%)
1	friend (i)	87
2	参照 (h)	94
3	演算子のユーザ定義 (f)	90
4	関数のオーバーローディング (g)	94
5	コンストラクタとデストラクタ (a)	97
6	インライン関数 (b)	100
7	const 関数 (e)	90
8	private, public, protected (c)	100
9	new と delete (d)	97
10	メンバ関数 (j)	100
11	デフォルトの引数 (k)	94

表4: C++の学習項目の困難さの順序の検定結果



一方、オブジェクト指向の基本的な概念として列挙した項目 (問9B) については、明確な困難さの順序が得られなかった。表5のように順序を付けたが、検定の結果、項目の順位に有意な差がほとんど見られなかった。ただし、多態性と他のいくつかの項目との間には有意な差が見られ、多態性が比較的難しいと認識されていることがわかった。

表5: オブジェクト指向の学習項目の困難さの順序

順位	学習項目	認知度 (%)
1	多態性 (c)	71
2	委譲 (f)	32
3	集約 (d)	65
4	継承 (e)	97
5	クラスとインスタンス (a)	100
6	カプセル化 (b)	100

3.4 学習を支援すべき項目

前述のように、各項目の学習の支援の必要性は、習得の困難さと習得後の有用性との両特質に依存すると考えられる。習得の困難さについての調査結果は、前節で述べた。本節では、習得後の有用性 (問9C, 問10C)、および、困難さと有用性との比較について述べる。

C++の学習項目の習得後の有用性 (問10C) については、習得の困難さ程明確ではないが、順序を付けられることがわかった。表6にその順序を、表7にその検定結果を示す。private, public, protected, コンストラクタ, デストラクタなどが比較的有用であると認識されており、friend が比較的有用でないとして認識されていることがわかる。

表6: C++の学習項目の有用性の順序

順位	学習項目
1	private, public, protected (c)
2	メンバ関数 (j)
3	コンストラクタとデストラクタ (a)
4	new と delete (d)
5	関数のオーバーローディング (g)
6	デフォルトの引数 (k)
7	const 関数 (e)
8	参照 (h)
9	インライン関数 (b)
10	演算子のユーザ定義 (f)
11	friend (i)

オブジェクト指向の学習項目の有用性 (問9C) については、表8のように順序を付けられ、検定の結果、ほとんどの項目の対について順位に有意な差があった。

C++の学習項目の困難さと有用性との関係を図1に示す。横軸は表6の有用性の順位に対応し、縦軸は表3の困難さの順位に対応している。有用で、かつ、難しいと明らかに認識されている項目は見当たらない。従って、比較的有用で、かつ、やや難しいと認識されている項目、あるいは、比較的有用ではな

表 7: C++の学習項目の有用性の順序の検定結果

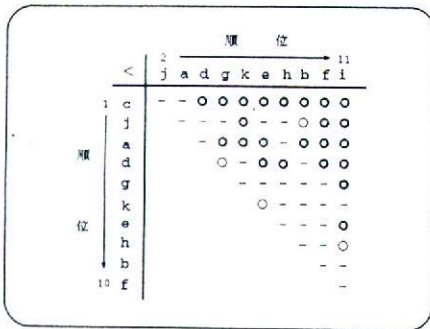


表 8: オブジェクト指向の学習項目の有用性の順序

順位	学習項目
1	カプセル化 (b)
2	クラスとインスタンス (a)
3	継承 (e)
4	多態性 (c)
5	集約 (d)
6	委譲 (f)

いが、難しいと認識されている項目が、学習を支援すべき項目の候補であると考えられる。例えば、friend(i)、演算子のユーザ定義 (f)、参照 (h)、関数のオーバーローディング (g)、コンストラクタとデストラクタ (a)、private、public、protected(c)である。ただし、学習者の認識が必ずしも正しいと限らないので、注意して頂きたい。学習者が認識していなくても有用な項目は存在する可能性がある。上の項目は、学習を支援すべき候補に過ぎない。

オブジェクト指向の学習項目については、前節で述べたように、習得の困難さの順位に有意な差が見られなかったため、困難さと有用性との比較は行わなかった。

3.5 支援の方法について

3.5.1 演習コースにおける問題点

回答者に提示した項目 (問 11) については、表 9 のように順序を付けることができたが、検定の結果、項目の順位に有意な差がほとんど見られなかった。下位の 2 項目と他の項目との間に有意な差が見られただけであった。人によって問題と思う点や学習の方法が大きく異なることが伺える。

列举した候補の他に、回答者自身が書き加えた項目には、学習時間の不足 (7 人)、技能の大きく異なる者が同一の演習コースを受けること (1 人)、C++が言語として洗練されていないこと (1 人) があった。

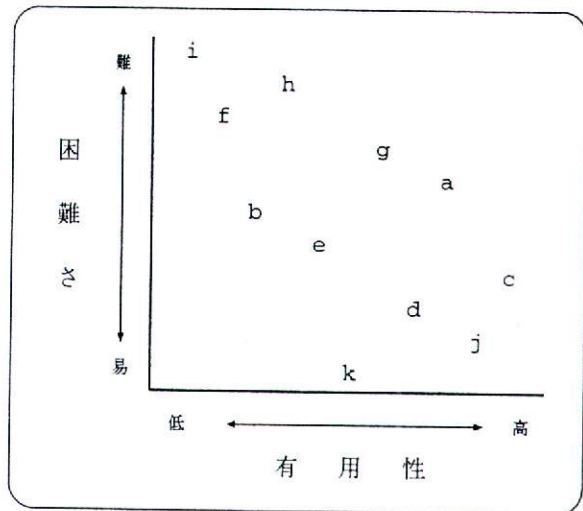


図 1: 習得の困難さと習得後の有用性との関係

表 9: 演習コースにおける問題点

- 1 C++に関するよい本がみつからなかった。(a)
- 2 疑問を聞ける人が身近にいなかった。(f)
- 3 よいサンプルプログラムがみつからなかった。(g)
- 4 OOに関するよい本がみつからなかった。(b)
- 5 提出物に十分なコメントが得られなかった。(h)
- 6 コンパイラの表示が不親切だった。(e)
- 7 C言語の理解が不足していた。(c)
- 8 演習問題が適当でなかった。(d)

3.5.2 演習コースにおいて効果的であった作業

列举した候補 (問 8) については、表 10 のように順序を付けられた。ただし、上位の 3 項目のそれぞれと他の項目との間で順位に有意な差が見られたが、その他の項目の対には有意な差が見られなかった。やはり、人によって学習の方法が大きく異なることが伺えるが、傾向としては、自身でプログラムを書いたり、本を読んだり、サンプルプログラムを読んだりしたことの効果が高く認識されていることがわかる。講師や他の参加者から説明を受けたり情報交換を行うよりも、各自が取り組んだ作業が効果的であると認識されていることが伺える。

列举した候補の他に、回答者自身が書き加えた項目には、OMTの説明を聞いたこと (1 人)、他人の演習の結果 (設計図) を見たこと (1 人) があった。

なお、演習コースの前に C++を学習した経験のある回答者とそうでない回答者との間では、興味深い差異が見られた。スライドで文法の説明を聞いたことや、CとC++の比較を聞いたことは、経験の

表 10: 演習コースにおいて効果的であった作業

1	プログラムを書いていた時の試行錯誤 (e)
2	C++の本を読んだこと (d)
3	サンプルプログラムを読んだこと (g)
4	CとC++の比較を聞いたこと (b)
5	文法の説明を聞いたこと (a)
6	OMTの本を読んだこと (c)
7	参加者の間での話 (h)
8	講師との会話やメールのやりとり (i)
9	演習の提出物にコメントをもらったこと (f)

ある者のグループ (問 7B で 2 または 3 と答えた 20 人) で比較的下位であった。ところが、経験のない者のグループ (問 7B で 1 と回答した 11 人) では、上位であった。表に 11 に、両グループにおける順位を対比する。学習の進捗に応じて、学習や学習支援の方法を変える必要性が伺える。

表 11: C++の経験者、未経験者の効果的な作業

	C++の経験者	C++の未経験者
1	プログラムを書いていた時の試行錯誤 (e)	プログラムを書いていた時の試行錯誤 (e)
2	サンプルプログラムを読んだこと (g)	文法の説明を聞いたこと (a)
3	C++の本を読んだこと (d)	CとC++の比較を聞いたこと (b)
4	OMTの本を読んだこと (c)	C++の本を読んだこと (d)
5	参加者の間での話 (h)	サンプルプログラムを読んだこと (g)
6	演習の提出物にコメントをもらったこと (f)	講師との会話やメールのやりとり (i)
7	講師との会話やメールのやりとり (i)	OMTの本を読んだこと (c)
8	CとC++の比較を聞いたこと (b)	参加者の間での話 (h)
9	文法の説明を聞いたこと (a)	演習の提出物にコメントをもらったこと (f)

3.5.3 学習に役立つツール

列挙した候補 (問 12) については、表 12 のように順序を付けられた。ただし、下位の 2 項目と他の項目との間で順位に有意な差が見られたが、その他の項目の対に有意な差がほとんど見られなかった。なお、ここで言う FAQ (frequently-asked questions) のデータベースとは、実際に質問された疑問点とその説明とを蓄積するデータベースである。プログラムのブラウザとは、プログラム中に定義したクラスや関数を図式で一覧表示してくれるツールである。

以上の結果と、3.5.1 と 3.5.2 の結果とを比較す

表 12: 役立つと思われる支援ツール

1	サンプルプログラムのデータベース (g)
2	FAQのデータベース (e)
3	プログラムのブラウザ (c)
4	標準関数を検索してくれるツール (d)
5	書いたプログラムを評価してくれるシステム (f)
6	出題し採点してくれる対話的システム (b)
7	電子化された教科書 (a)

ると、次の 2 点が興味深い。1 点目は、よいサンプルプログラムが見つからなかったこと、サンプルプログラムを読んだこと、サンプルプログラムのデータベースは、どれも上位にあることである。サンプルプログラムを読むことの重要性が伺える。2 点目は、本に対する関心が高く、そして、本を読んだことの効果は認識されている一方で、電子化された教科書が比較的役立たないと認識されていることである。本は必要であっても、現状では、紙に印刷されたものの方が、電子化されたものよりも優れていると認識されていると思われる。

3.6 調査結果のまとめ

主な結果をまとめると次のようになる。

各学習項目の困難さについて C++の基本的な概念のいくつかについて、その困難さに表 3 のような比較的明確な順序が付けられる。

学習を支援すべき項目について C++の基本的な概念のいくつかについて、その困難さと有用性との間に図 1 のような関係が見られる。その関係から、friend、演算子のユーザ定義、参照、関数のオーバーローディング、コンストラクタとデストラクタ、private、public、protected などが、学習を支援すべき項目の候補と考えられる。

支援の方法について 学習の方法は人によって大きく異なると思われるが、自身でプログラムを書いたり、本を読んだり、サンプルプログラムを読んだりすることの効果が高く認識されている傾向がある (表 10)。従って、そのような作業を支援することが効果的であると考えられる。特に、サンプルプログラムのデータベースに対する要望は高い傾向がある (表 12)。なお、本を読むことの効果が高く認識されている一方で、単に電子化された教科書に対する要望は低い。

その他 習得に必要と思われる日数に対して習得に費やしてもよいと思われる日数が短い傾向がある。

なお、本調査では、企業の技術者と学生との間で

認識に大きな差異が見られなかった。

4 おわりに

C++の学習の問題点についてのアンケート調査を実施し、3.6にまとめたような情報が得られた。C++の学習を支援する上で重要な指針が得られたと考えている。ただし、本調査では、学習者を対象としたので、現状の学習者の認識はよくわかったが、教育者の観点や新しい支援方法などの情報が直接的には得られていない。それらの情報を収集することは今後の重要な課題である。また、今回は31名の回答を分析したが、より詳細な情報を得るためには、更に多くの回答を得る必要があると思われる。調査を継続して、多数の回答を得ることも今後の重要な課題である。

今後は、調査結果に基づいて、学習支援の具体的な方法についても検討する予定である。

謝辞

時間を割いてアンケートに回答して頂いた31名の方々に心より感謝する。

参考文献

- [1] B. Stroustrup, The C++ programming language, Addison-Wesley, 1994.
- [2] E. F. Gehringer, and M. L. Manns, "OOA / OOD / OOP: What programmers and managers believe we should teach," Journal of Object-Oriented Programming, vol. 9, no. 6, pp. 52-60, Oct. 1996.
- [3] J. Rumbaugh, Object-Oriented Modeling and Design, Prentice Hall, 1990.
- [4] K. Lato and A. Drechsler, "Effective Training in OOT-Learn by doing," Journal of Object-Oriented Programming, vol. 9, no. 6, pp. 48-51 63, Oct. 1996.
- [5] 増永 良文, 永田 守男, "オブジェクト指向技術とその応用論文特集の発行にあたって," 信学誌, J79-D-I, no. 9, pp. 579-584, Oct. 1996.
- [6] 南風原, "III-28 符合検定," 統計ガイドブック, 池田央, p. 150, 新曜社, 1989.
- [7] 山崎 利治, "共通問題によるプログラム設計技法解説," 情報処理, 25, no. 9, p. 934, 1984.

付録: 実際のアンケート

1. 現在、主にどんな計算機のプログラムを作っていますか?
次の5つから当てはまるもの全てを選択して下さい。
 - (1) 大型計算機、中型計算機
 - (2) UNIX系ワークステーション
 - (3) パソコン
 - (4) 組み込みプロセッサ
 - (5) その他 ()
2. プログラムを新規に書く場合は、普段、どのようなスタイルをとっていますか?
A、Bについて、1つずつ選択して下さい。
 - A. (1) いきなり計算機に向かう。
(2) 机上でよく設計してから計算機に向かう。
 - B. (1) プログラムを全部書いた後で初めてコンパイルする。
(2) プログラミング途中で何度もコンパイルする。
3. 本演習を受ける前は、どんなプログラミング言語を学んだことがありますか? よく知っている順に挙げて下さい。たくさんある場合は、5つまで挙げて下さい。また、それぞれの言語について、書いたことのあるプログラムのおよその大きさ(他から流用した部分を除く)を次の中から1つを選択して下さい。
 - (1) その言語でプログラムを書いたことがない。
 - (2) 1行以上 100行未満
 - (3) 100行以上 1万行未満
 - (4) 1万行以上
4. 現在、C言語ではどの程度プログラムを書けますか?
次の3つから1つを選択して下さい。
 - (1) 本がないと全くプログラミングができない。
(標準関数のオンラインマニュアルは見てもよい)
 - (2) 時々本を見ればプログラミングができる。
 - (3) 本を見なくてもプログラミングができる。
5. 本演習を受ける前は、次の概念をどの程度使っていましたか?
 1. ポインタ
 2. 構造体(struct)
 3. ソースファイルの分割(分割コンパイル)
 4. 抽象データ型
 5. ヘッダファイル(.hファイル)それぞれ、次の中から1つを選択して下さい。
 - (1) 言葉も知らなかった。
 - (2) ほとんど使えなかった。
 - (3) 少しは使っていた。
 - (4) よく使っていた。
6. 現在、C++の環境としては主に何を使っていますか? 次の中から当てはまるもの全てを選択して下さい。
 - (1) UNIX上のコンパイラやデバッガ(g++, gdbなど)
 - (2) Microsoft Visual C++
 - (3) パソコン上で使えるその他の環境
(Borland C++, Symantec C++など)
 - (4) その他 ()
7. 本演習を受ける前に、オブジェクト指向やC++を使ったことがありますか? それぞれについて、次の3つから1つを選択して下さい。
(ここで言う「オブジェクト指向を使う」とは、分析・設計を行ったり、データベースを使う時にオブジェクト

指向的な考え方をすることです。)

- A. オブジェクト指向
- (1) 学んだことも使ったこともない。
 - (2) 講義や講習会などに出席したり、本を読んだことはあるが、実際には使ったことがない。
 - (3) 練習問題では使ったことがある。
- B. C++
- (1) 学んだことも使ったこともない。
 - (2) 講義や講習会などに出席したり、本を読んだことはあるが、実際に使ったことがない。
 - (3) 練習問題では使ったことがある。

以降の8~12の質問への回答では、項目を選択するのではなく、項目を並べ替えて頂きます。特に指定のない限り、項目を省かずに、必ず、項目の全てを並べ替えて下さい。

8. 本演習の中で何をすることが学習に役立ったと思われるか? 次の項目の中から実際に行ったことを選び出し、役立ったと思われるものから役立たなかったものへ順に並べ替えて下さい。少しでも行ったことは、役立たなかったとしても、必ずお書き下さい。

- (a) 文法の(スライドでの)説明を聞いたこと
- (b) CとC++の(スライドでの)比較を聞いたこと
- (c) OMTの本を読んだこと
- (d) C++の本を読んだこと
- (e) プログラムを書いていた時の試行錯誤
- (f) 演習の提出物に対するコメントをもらったこと
- (g) サンプルのプログラムを読んだこと
- (h) 参加者の間での話
- (i) 講師との会話やメールのやりとり
- (j) その他 ()

9. オブジェクト指向のいくつかの概念について、次のA,B,Cの質問にお答え下さい。

A. 現在、次の(a)~(f)の概念の中でどれを知っていますか? 言葉は聞いたことがあるというものを全て挙げて下さい。

- (a) クラスとインスタンス
- (b) カプセル化
- (c) 多態性 (Polymorphism)
- (d) 集約 (Aggregation)
- (e) 継承 (Inheritance)
- (f) 委譲 (Delegation)

B. Aで選択された概念を、御自身が誰かに教えることを想像して下さい。どれの説明が難しいですか? 説明しにくいと思われる順に並べ替えて下さい。

C. Aで選択された概念は、プログラミングを行う上で役立つと思えますか? より役立つと思われる順に並べ替えて下さい。

10. C++について、次のA,B,Cの質問にお答え下さい。

A. 現在、次の(a)~(k)の概念の中でどれを知っていますか? 言葉は聞いたことがあるというものを全て挙げて下さい。

- (a) コンストラクタとデストラクタ
- (b) インライン関数
- (c) private, public, protected
- (d) new と delete
- (e) const 関数

- (f) 演算子のユーザ定義
- (g) 関数のオーバーローディング
- (h) 参照
- (i) friend
- (j) メンバ関数
- (k) デフォルトの引数

B. Aで選択された概念を、御自身が誰かに教えることを想像して下さい。どれの説明が難しいですか? 説明しにくいと思われる順に並べ替えて下さい。

C. Aで選択された概念はプログラミングを行う上で役立つと思えますか? 役立つと思われるものから役立たないと思われるものへ順に並べ替えて下さい。

11. C++の学習の過程では何が問題でしたか? 次の項目を、問題であったものから問題でなかったものへ順に並び替えて下さい。

- (a) C++に関するよい本がみつからなかった。
- (b) オブジェクト指向に関するよい本がみつからなかった。
- (c) C言語の理解が不足していた。
- (d) 演習問題が適当でなかった。
- (e) コンパイラの表示が不親切だった。
- (f) 疑問があった時に聞ける人が身近にいなかった。
- (g) よいサンプルプログラムがみつからなかった。
- (h) 提出物に対して十分なコメントが得られなかった。
- (i) その他 ()

12. どのようなツールがあれば学習に役立ったと思えますか? あればよかったと思われるものから、なくてもよかったと思われるものへ順に並び替えて下さい。

- (a) 電子化されたオブジェクト指向やC++の教科書 (CD-ROMのようなもの)
- (b) 自動的に問題を出し、回答を採点してくれるシステム (オブジェクト指向やC++の概念について)
- (c) プログラムのブラウザ (定義したクラスなどを一覧から選べるようなグラフィカルな環境)
- (d) 標準関数を検索してくれるツール (単なるオンラインマニュアルでないもの)
- (e) FAQ (Frequently Asked Question) のデータベース (よく出る質問とその回答との一覧)
- (f) プログラムの評価システム (書いたものを自動的に解析して、評価を返したり問題点を指摘する)
- (g) サンプルプログラムのデータベース (検索の機能を持つもの)
- (h) その他 ()

13. C++の学習だけに時間を費やせたとすれば、どのくらいの期間でC++を習得できたと思えますか? 日数をお書き下さい。

14. C++の習得によって得られたこと(これから得られるかもしれないこと)を考えるとどのくらいの期間であれば、C++の学習に費やしてよかったと思えますか? のべ日数をお書き下さい。

15. オブジェクト指向やC++やプログラム言語の学習に関して、その他の御意見や御感想を御自由にお書き下さい。