

今回の内容

3.1 プログラミング言語とは . . . . .	3-1
3.2 プログラミング言語 C . . . . .	3-3
3.3 演習問題 . . . . .	3-6
3.4 オフィスアワーについて . . . . .	3-9

今回の実習から、C というプログラミング言語を用いた、プログラミングの入門を行います。

### 3.1 プログラミング言語とは

いろいろな用途にコンピュータが使われています。複雑な計算を行ったり、携帯電話を制御したり、ビデオゲームを楽しませてくれたりします。しかし、単にコンピュータという物(ハードウェア)があっただけでは、それは何の役にも立ちません。コンピュータが役に立つのは、そのコンピュータがどのように働くべきかを指示したプログラム(ソフトウェア)が用意されているからです。コンピュータの行うべき仕事を事細く記述したものを「コンピュータプログラム」、あるいは単に「プログラム」と呼びます。このとき、どのようなプログラムを書けばどのようにコンピュータが働くかについての約束事が必要になります。プログラムの書き方とその解釈のされ方があらかじめ決っていなければ、コンピュータに意図した通りの働きをさせることはできません。その「約束事」を提供してくれるものをプログラミング言語と呼びます。日本語が、日本語を話す人の間での意思疎通を可能にしてくれるのと同様に、1つのプログラミング言語は人間の意図をコンピュータに伝えることを可能にしてくれます。

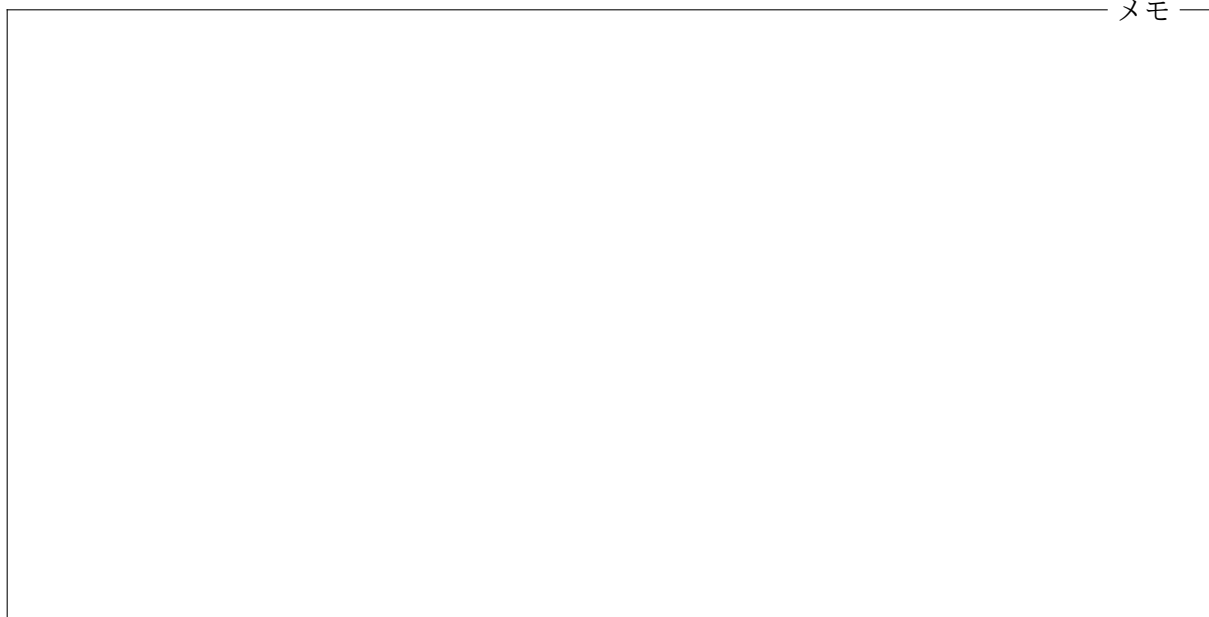
現在広く利用されているコンピュータは、その中に「メモリ」と呼ばれる記憶装置を備えており、ここにいろいろな情報を保存することができるようになっています。見掛けの上では、メモリに蓄えられる情報は0と1の二種類の数だけからなる長い長い数列のように見えます。人間が扱う普通の文章や、数値などの情報は、適当な規則にしたがって、0と1からなる数列として表現されてメモリ中に記憶されます。コンピュータ自身の働きを指定するためのプログラムもやはりこのメモリ中に置かれます。0と1の数個から数十個の並びを1つの単位として、コンピュータが行うべき作業を指示します。どのような並びでどのような作業を行うかについての約束事は、そのコンピュータによってあらかじめ決められています。コンピュータはこの約束事にしたがって順に指示されたとおりの作業を行っていきます。この指示の単位となる0と1の並びを機械語命令と呼びます。1つ1つの機械語命令は、非常に単純な作業しか行わせることはできませんが、この単純な作業を何千、何万と組み合わせることにより、コンピュータに非常に複雑な作業を行わせることができます。機械語命令をたくさん組み合わせて、コンピュータの行うべき一連の作業を指定したものを機械語プログラムと呼びます。

メモ

コンピュータに一連の作業をさせるには、この機械語プログラムというものを用意することが必要ですが、1つ1つの機械語命令がどのような作業に対応するかについての約束事は一般に非常に複雑であり、また、1つ1つの機械語命令は非常に単純な作業しか行わせることができないため、ある程度のまとまった作業をコンピュータに行わせるために必要な機械語プログラムを人間が直接作るのは相当に困難です。そこで、もっと人間が理解しやすい書き方でコンピュータの行うべき仕事を記述し、それを機械語プログラムに機械的に変換することでコンピュータに仕事をさせるという方法がよく用いられます。この、より人間が理解しやすい書き方も、一定の約束事(どう書けば、どう働くか)に基づいて書かれますが、ちょうど、人間の日常の言語に英語や日本語、中国語の違いがあるように、この「より人間が理解しやすい書き方」にも、その用途に応じていろいろなものがあります。機械語に対して、より人間が理解しやすいプログラミング言語を総称して、一般に「高級(プログラミング)言語」と呼びます。ここでの「高級」という語は「優れている」という意味ではありませんので誤解しないでください。

高級プログラミング言語はコンピュータが直接理解することはできませんから、何らかの方法でコンピュータの理解できる機械語へ変換してやらなければなりません。これには大きく分けて2つの方法があります。その1つは、高級言語で書かれたプログラムを少しずつ読み取りながら、そこで指示されている作業を、機械語でコンピューターに指示し、実行させて行く方法です。あたかもコンピュータに対して人間の話している言葉をその場で通訳しているかのように働きますので、このような仕組みはインタプリタ(通訳)と呼ばれます。通常、このインタプリタ自身も1つのコンピュータプログラムとして実現されます。もう1つの方法は、コンピュータがそのプログラムの実行を始める前に、高級言語で書かれたプログラムをあらかじめ機械語にすべて翻訳しておく方法です。一旦翻訳が済んでしまえば、翻訳済みの機械語プログラムは何度でも利用することができます。この翻訳作業のことをコンパイルと呼び、翻訳を行ってくれる仕組みをコンパイラと呼びます。また、コンパイルする前の高級言語で書かれたプログラムをソースプログラムと呼び、コンパイルして得られた機械語プログラムのことをオブジェクトプログラムと呼びます。コンパイラ自身もやはり1つのコンピュータプログラムとして実現されます。

メモ



## 3.2 プログラミング言語 C

C は高級プログラミング言語の1つです。コンピュータの機械語を知らなくても、コンピュータに意図した働きをさせることができます。C は通常コンパイラによる方法で利用されます。非常に単純なプログラムを例として、プログラムの作成と、コンパイル、コンパイルして出来上がったオブジェクトプログラムの実行を行ってみましょう。

**C プログラムの作成** コンピュータの記憶装置に格納されたプログラムやデータなどの個々のまとまりを「ファイル」と呼びます。ソースプログラムもオブジェクトプログラムもコンピュータの中では、1つのファイルとして存在することになります。

C のソースプログラムは、アルファベットや数字、記号などの文字が並んでできあがったものですから、ワープロで文書を編集するのと同じようにして作ることができます。ただし、ワープロの場合は、文字の大きさ、太さ、飾りなどを含めた文書の見掛けが重要ですが、C のソースプログラムとしては、単にそこにどんな文字がどの順に並んでいるかだけが重要になります。このような「文字の並び」としてだけの情報を持ったファイルは、一般に「(プレーン) テキストファイル」と呼ばれます。C のソースプログラムは単に、このテキストファイルの1つにしか過ぎません。

Windows 環境などのパソコンの環境では、それぞれのファイルに名前を付けることができ、その名前でファイルを区別することができますが、C のソースプログラムが書いてあるファイル名は、通常、末尾を「.c」<sup>1</sup>とします。ここでは、myfirst.c という名前のファイルに、次のようなソースプログラムを書き込んでみましょう。

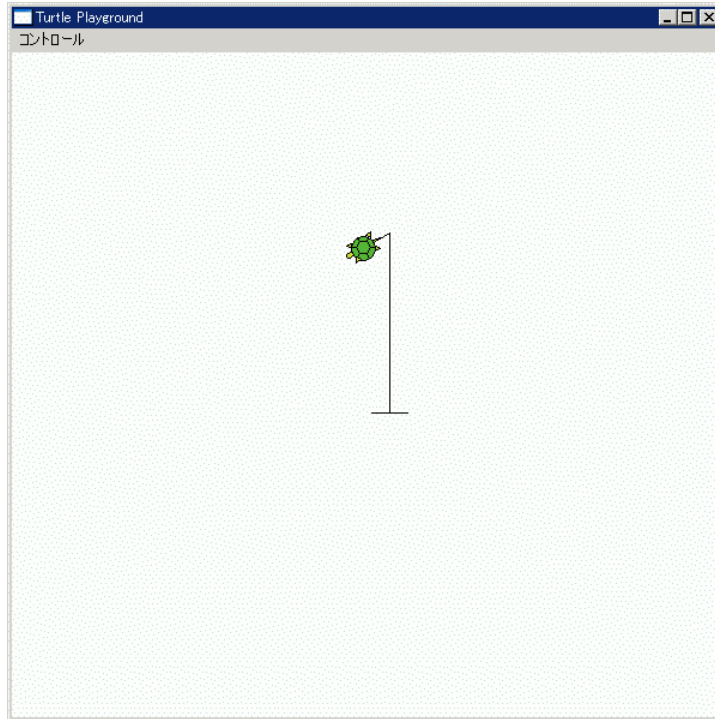
```
myfirst.c
#include <turtle.h>

main()
{
    tForward(30);
    tBackward(15);
    tTurn(90);
    tForward(150);
    tTurn(120);
    tForward(28);
}
```

この C プログラムをコンパイルして、パソコンで実行すると、次のようなウィンドウが現れ、その中のカメが動いて、数字の1の形を描いてくれます。



<sup>1</sup>このように、ファイルの種別を区別するために、ファイル名の末尾につけられた「.」で始まる文字列を、Windows 環境では「拡張子」と呼びます。



Windows 環境では、TeraPad やメモ帳などといった「テキストエディタ」と総称されるソフトウェア (これもプログラム) を用いてテキストファイルを編集することができますので、これを用いてソースプログラムを書きます。C のソースプログラムとしては、どのような文字がどのように並んでいるかだけが意味を持ちますから、どんなテキストエディタを使って書いたとしても、できあがったテキストファイル中の文字の並びが同じである限り、そのプログラムの働きには影響はありません。

使用しているコンピュータの設定によっては「\」という文字が「¥」と表示されたり、キーボードに「\」という文字のキーの代わりに「¥」の文字があったりしますが、この2つの文字の違いを気にする必要はありません。キーボードのキーを直接叩いて入力された文字ならどちらでも同じようにプログラムは実行されます。ただし、かな漢字変換システム等によって入力された「¥」の文字 (通常は幅広に表示されます) は別の文字として認識されますので「\」と同じ意味に使うことはできません。

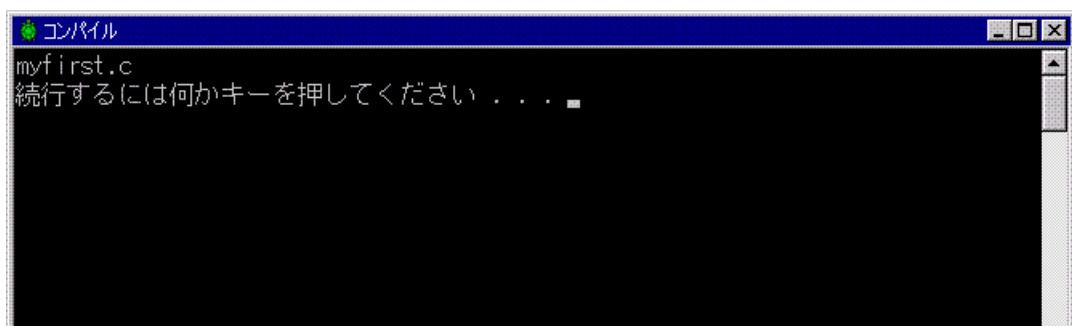
メモ

**C プログラムのコンパイル** ソースプログラムをファイル (`myfirst.c`) に保存したら、そのソースプログラムをコンパイラを使ってコンパイルし、オブジェクトプログラムに変換します。コンパイラ自身も1つのオブジェクトプログラムですから、アプリケーションプログラムの一つとして、それを起動することになります。実習室の Windows 環境にも、C 言語のコンパイラがいくつかイ

インストールされていますが、この「計算機基礎実習 I」ではという科目では、これらのコンパイラの内の1つを簡単に使うために、デスクトップにショートカットを用意して、それを用いて C プログラムのコンパイルを行います。このショートカットは、デスクトップ上に次のようなアイコンで表示されるはずですが。



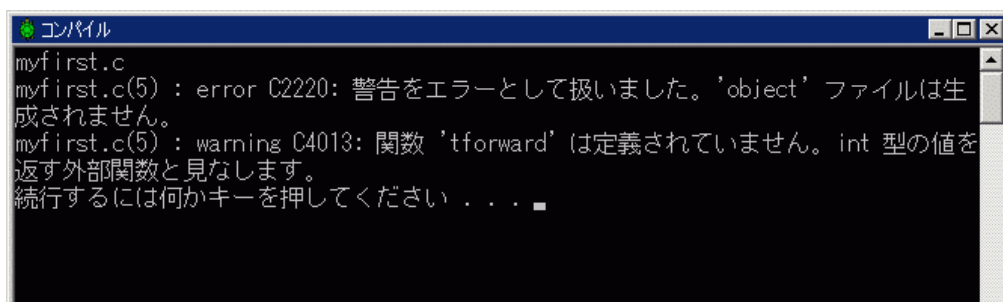
先ほどテキストエディタで作った `myfirst.c` という C プログラムをコンパイルするためには、この `myfirst.c` というファイルを、このアイコンへドラッグ・アンド・ドロップします。こうすると、コンソールウィンドウが現れ、正常にコンパイルできた場合には次のように表示されるはずですが。



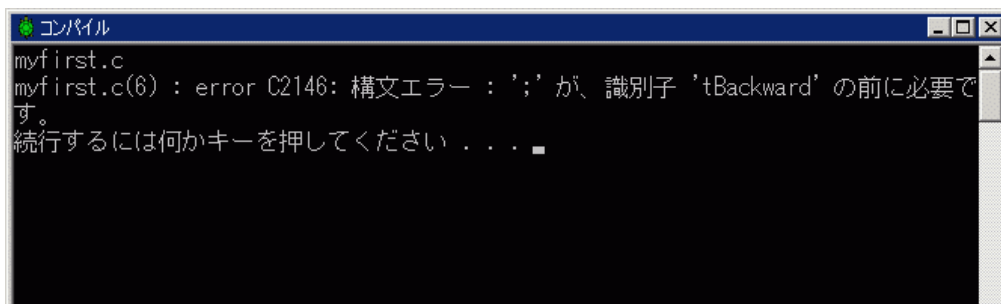
このコンソールウィンドウは、Enter キーを押せば消えてなくなります。正常にコンパイルできていれば、ソースファイル `myfirst.c` のあったフォルダ (ディレクトリ) に `myfirst.exe` という名前のオブジェクトプログラムが作成されます。Windows 環境の設定次第で「.c」や「.exe」などの拡張子は表示されないので注意してください。コンパイルしてできたオブジェクトプログラム、次のようなアイコンで表示されます。



もし、ソースプログラムに何らかの間違いがあって、正常にコンパイルができなかった場合、コンソールウィンドウには、その間違いの内容が表示されます。この場合 `myfirst.exe` というオブジェクトプログラムは作成されません。次は、ソースプログラム `myfirst.c` 中の「`tForward(30);`」の部分で「`tforward();`」と書き間違えてコンパイルに失敗した場合の例です。



また「tForward(30);」の「;」を書き忘れると次のようになります。



```
コンパイル
myfirst.c
myfirst.c(6) : error C2146: 構文エラー : ';' が、識別子 'tBackward' の前に必要です。
続行するには何かキーを押してください . . .
```

何にせよ、正常にコンパイルができない場合は、テキストエディタで myfirst.c の内容を修正し、もう一度、コンパイルを行います。

メモ

**オブジェクトプログラムの実行** コンパイルが成功したら、いよいよできあがったオブジェクトプログラムの実行です。オブジェクトプログラムは、そのアイコンをダブルクリックすることで、実行することができます。

ソースプログラムが正常にコンパイルできた場合でも、できあがったオブジェクトプログラムが期待した通りに動作しない場合もあります。このような場合もやはり、テキストエディタでソースプログラム myfirst.c の内容を修正し、もう一度コンパイルし直すことになります。一方、正常に動作するオブジェクトプログラムが完成してしまえば、そのオブジェクトプログラムは何度でも実行することができますので、実行の度にコンパイルしてやる必要はありません。

メモ

### 3.3 演習問題

1. 例として挙げた myfirst.c というプログラムをテキストエディタを使って作成し、コンパイル、実行してみなさい。

2. 次のような内容の C プログラムを作成、コンパイル、実行してみなさい。プログラムの内容を理解する必要はありません。ただし、ソースプログラムのファイル名は `fakelogo.c` としなさい。

```
fakelogo.c
#include <turtle.h>

main()
{
    int i;

    tSetColor(0.4,0.0,0.4);
    for (i = 0; i < 4; i++) {
        tTurn((3-i/2)*45);
        tForward(8*(7-(i/2)*(2+(i/3)*3)+(i%2)*12));
    }
    tFill();
    tTurn(-90);
    tSetColor(0.0,0.4,0.0);
    tForward(50);
    for (i = 0; i < 4; i++) {
        tForward(40+60*(i%2));
        tTurn(90);
    }
    tFill();
    tSetColor(0.0,0.3,0.5);
    for (i = 0; i < 4; i++) {
        tTurn(-45-(i%2)*90);
        tForward(158-(i%2)*108);
    }
    tFill();
}
}
```

## 今回の実習内容

基本的には自分のペースで進めてください。課題3と課題4、課題5については教員(中野)が説明します。これらの課題には想定されている開始時刻が書かれています。説明が終わったら、また自分のペースで進めてください。

分からないところがあったら、手をあげて TA さんと呼んでください。再履修の学生さんもすべての課題に取り組んでください。

1. 著作権情報センターの「みんなのための著作権教室」や「著作権 Q&A」の Web ページで「著作権」について勉強してください。

「学ぼう著作権」や「こんな時の著作権」、「デジタル・ネットワーク社会と著作権」、「学校教育と著作権」のコーナーを中心に自分の知識を確認してください(第3回実習の Web ページにリンクがあります)。

2. 前回の課題6で提出したレポートを見てみましょう。

レポート一覧の Web ページ(第3回実習の Web ページにリンクがあります)から他の受講者のレポートを見ることができます。自分のレポートと比較してみましょう。レポートは学内のみの公開です。

3. 他の受講者のレポートを採点してください(16:00)。

次の手順にしたがって、他の受講者 10 名のレポートを採点してください。

- (1) 「課題の提出と確認」の Web ページを開きます。
- (2) 「課題」の右のメニューから「第 3 回 前回のレポートの採点」を選択します。
- (3) 「送信」のボタンをクリックします。「提出するファイル」は空欄のまま構いません。
- (4) 10 名分のレポートを採点するページが表示されますので、それぞれのレポートを 10 点満点で採点して「採点結果を送る」のボタンをクリックしてください。

採点を行う際には、次の点を評価に取り入れてください。

- 提出者自身の言葉でまとめてありますか? Web ページ中の文章の言い回しを変えただけの部分がありませんか? (重要度★★★)
- レポートの内容は、それぞれの課題について分かりやすくまとめられていますか? 読んだ人に参考になる情報が含まれていますか? (重要度★★★)
- 多くの Web ページから情報が集められていますか? 特定の Web ページの情報だけに基づいたレポートになっていませんか? (重要度★★)
- 参考にした Web ページのアドレス (URL) が明記されていますか? (重要度★)
- 電子メールの件名 (Subject) はありますか? (重要度★)

この他にも、各レポートの良いところ、悪いところを点数に反映して頂いて構いません。ごく平均的と思われるレポートが 5 点となるような採点を心掛け、できるだけ、レポートの点数に差をつけるようにしてください。悪いレポートには、思いきって 0 点を、すばらしいレポートには満点をあげてください。

前回の課題 6 の点数は、そのレポートを採点した人 (10 人程度) がつけた点数の平均点 (小数点以下四捨五入) となります。また、この採点の作業自体も 10 点満点で (機械的に) 評価されますので、充分時間をかけて慎重に評価してください。採点は何度でもやり直すことができ、最後に送られたものがあなたの採点として採用されます。あなたの採点が、他の採点者を含めた平均点に近ければ近いほど、また、あなたが、レポートの点数に差をつければつけるほど、この課題 (レポートの採点) に対する評価は良くなります。この課題の提出の締め切りは 5 月 11 日 (木) 13:00 までです。レポートは学内のみの公開ですので注意してください。

4. このプリントの「プログラミング言語とは」と「プログラミング言語 C」の節の内容を担当教員が解説します (16:45)。

5. プログラミングの準備をします。「授業共有フォルダ (R:)」の a89023 というフォルダの中の「計算機基礎実習 I」というフォルダに置いてある「コンパイル」と「TeraPad」という 2 つのファイル (ショートカット) をデスクトップにコピーしてください。プログラミングの勉強をするための準備はこれだけです。以下で説明する「パス名」という言葉を覚えておいてください。



パス名 コンピュータの中のファイルやフォルダを指し示すときに、それが格納されているドライブ名から始めて、目的のファイルに辿り着くまでのフォルダの名前を「\」で区切って書き、最後に目的のファイルやフォルダの名前を書き記した「パス名」と呼ばれる書き方をすることがあります。たとえば、上でコピーした「コンパイル」というファイルのパス名は次のようになります。

R:\a89023\計算機基礎実習I\コンパイル.lnk

このパス名の末尾に現れている「.lnk」は、このファイルの種類を表す「拡張子」と呼ばれるもので、本来はファイル名(やパス名)の一部ですが、Windows 環境の設定によっては表示されない場合もあります(みなさんの環境では通常は表示されません)。

#### 6. C 言語のプログラムを、コンパイルして、実行してみましょう。

TeraPad を使って、このプリントのプログラム `myfirst.c` と `fakelogo.c` のソースファイルを作り、「ホームドライブ(Q:)」に保存し、コンパイルして、実行してみましょう。TeraPad は、スタートメニューから「プログラム」→「文書作成」→「TeraPad」を選択すると起動できます。プログラムの内容が全く同じになるようにしてください。特に、大文字と小文字、「.」と「,」の違いに注意しましょう。空白部分の長さは気にする必要はありません。

ソースファイルをデスクトップの「コンパイル」にドラッグ・アンド・ドロップすると、コンパイルされてオブジェクトファイルが作成されるはずです。作成されたオブジェクトファイルをダブルクリックすると実行することができます。うまくコンパイルと実行ができれば、ソースファイル(`myfirst.c` と `fakelogo.c`)をそれぞれ「課題の提出と確認」のページから提出してください。この課題の提出期限は5月12日(金) 15:20です。

#### 7. このプリントを読んで、クイズに答えてください。

口頭での説明内容を思い出しながら、ゆっくりこのプリントを読んで下さい。「課題の提出と確認」の Web ページで「第3回 クイズ」を選択し、「送信」のボタンをクリックしてクイズに答えてください。このクイズは何度でもやってみることができます。最高得点を評価の対象としますので、満点を狙ってください。ただし、Web ブラウザの「戻る」ボタンを使用することはできません。この課題の提出期限は5月12日(金) 15:20です。このクイズには課題1で勉強した著作権に関する問題も含まれています。

### 3.4 オフィスアワーについて

「計算機基礎実習 I」など、担当授業に関する質問や相談をして頂くために、教員(中野)が特定の場所に待機しておく「オフィスアワー」と呼ばれる時間帯を設けています。気軽に利用してください。2017年度前期の中野のオフィスアワーは次の通りです。

毎週 月曜2講時(1-514室)と木曜昼休み(1-614室)

この他にも「チューター制度」といって大学院生が分からないところを教えてくれる制度(詳しくは掲示物を見てください)もあります。

計算機基礎実習 I ・ 第3回 ・ 終わり