Table of Contents

Preface

CNC machining centers are among the most popular forms of CNC metal-cutting machine tools. Just about every company that uses metal-cutting CNC machines has at least one machining center:

It is the intention of this text to introduce beginners and experienced shop people alike to the programming techniques needed for utilizing CNC machining centers. We will begin in a basic manner, ensuring that even newcomers to the machine shop environment will be able to follow the presentations. And we use a building blocks approach – so as you get deeper into the material – we will be adding to what you already know. When you're finished, you will have a thorough understanding of what it takes to program a CNC machining center.

While we will discuss a few setup- and operation-related topics, setting up and running a CNC machining center are beyond the scope of this text. Indeed, they are perquisites to this text. If you wish to learn how to setup and run CNC machining centers, we do offer a separate text that will teach you these skills (to get the *CNC Machining Center Setup and Operation* manual, please uses our contact information on the back cover of this text).

We use a *Key Concepts* approach to presenting CNC. The Key Concepts allow us to minimize the number of major topics you must master in order to become proficient with CNC machining center usage. With this approach, there are six Key Concepts related to programming CNC machining centers.

Beginners should concentrate most on understanding the points made early in each Key Concept. It is as important to know *why* you are doing things as it is to know how to do them. Concentrate first on the whys. It will be impossible for beginners to totally memorize and comprehend every technique used with machining centers the very first time it is presented or read. Rest assured that if you can understand the basic reasoning behind why each CNC feature is required, it will be much easier to master the use of the feature. Once this basic reasoning is understood, it will be relatively easy to review the related material to extract specific details of how each CNC feature is used – so you can start putting your CNC machining center to good use.

For experienced CNC people, this text will provide you with alternatives. You'll be able to compare your current methods to our recommendations. If you have previous CNC machining center experience for other types of CNC controls, you will find it relatively easy to adapt what you already know to the most popular CNC control in the industry, the Fanuc control. If you have had experience with other types of CNC machine tools, such as turning centers or wire EDM machines, this text will help you adapt what you know to CNC machining centers.

As experienced people know, there are many ways to utilize CNC equipment. This text will show you one or two safe ways to accomplish your tasks. You can use your own common sense and past experience to develop your own style.

Prerequisites

Rest assured that this text will cover CNC machining center programming from the ground up. We will, however, assume that you possess some basic machining practice skills. And we will assume that you are acquainted with CNC machining center setup and operation. A CNC programmer must be able to create the documentation used to setup and run the workpieces for which they create programs. They must, of course, understand what it takes to setup and run a CNC machining center in order to do so. Again, these topics are addressed in a separate text: *CNC Machining Center Setup and Operation*, which is sold separately.

Basic machining practice experience

The more you know about basic machining practices, the easier it will be to learn how to program a CNC machining center. If, for example, you have worked on conventional machine tools in the machine shop, much of what you already know can be directly applied to CNC machining centers.

The more shop experience you have, the easier it will be to learn how to program CNC machining centers. If you have no previous shop experience, again, the needed topics are addressed in the manual entitled *CNC Machining Center Setup and Operation*.

Math

The word *numerical* in computer numerical control implies that numbers are highly involved with CNC. Indeed, CNC programmers will be making countless calculations every day. However, most calculations are quite simple to make. The types of arithmetic calculations required to create a typical CNC machining center program include addition, subtraction, multiplication, and division. Some programs do require more complex math, including right-angle trigonometry, as some programs require the use of trigonometry to develop. As with basic machining practices, the more you already know about math, the easier it will be to learn CNC programming.

Motivation

This should go without saying. We assume that you are motivated to learn. If you are highly motivated to learn about CNC machining centers, it will make your task much easier. Your motivation will help you overcome any obstacles you may have with learning the material in this text. With motivation, you’ll stick to it until you understand.

Controls covered

Since the Fanuc control is the most popular CNC control available, specific presentations in this text are provided in Fanuc format. Keep in mind, however, that the Key Concepts approach we use throughout this text will make it possible for you to learn techniques that can be applied to just about any CNC control in use today. Also keep in mind that several control manufacturers claim that their controls are *Fanuc compatible*. These manufacturers include Yasnac, Mitsubishi, Tasnac, Mazatrol (EIA option), Haas, Fadal, and Flashcut (among others). Even if you do not have one of these controls, we truly feel that if you understand the basic concepts, and if you understand how specific techniques are applied to one particular control type, it will be relatively easy to apply what you know to just about any CNC machining center control being used today.

Limitations

Please take note that our first goal will be to acquaint you with the usage of three axis machining centers (having X, Y, and Z axes). For the bulk of this text, only these three linear axes will be discussed. Certain machining centers do have additional axes, and will be beyond the scope of this text.

We will be discussing both vertical as well as horizontal machining centers throughout this text. Since vertical machining centers are easier to work with and visualize, and since vertical machining centers are much more popular than horizontals, most of the examples given will be for vertical machining centers.

This text will not specifically address five axis machining centers. Though many of the same principles apply, we do not wish to present five axis machines during this introductory text.

Programming method

This text teaches *manual programming* – also called G code level programming. While we’re listing this as a limitation, knowledge of G code is extremely important. With this method of programming, a programmer provides commands in the same “language” that the machine uses to execute it. While there are other ways to develop programs (using a computer aided manufacturing [CAM] system or a conversational CNC control) every CNC programmer must understand G code. CAM system programmers must, for example, be able to modify CNC programs at the machine when changes are required. These changes must be made at G code level.

The need for hands-on practice

We cannot overstress the need for practice on an actual machining center in order to fully master what you have learned in this text. If you are enrolled in a CNC course at a technical school, your instructor will probably provide hands-on practice with one of the machining centers in the school’s lab.

If you are reading this text on your own, you will not get this needed practice. We cannot, of course, provide the experience you would get from practicing on a real CNC machining center in this text. No text can provide such practice. We can, however, provide you with the principles and concepts that you must understand before you can spend any meaningful time at the machine.

Instruction method

This text is commonly used with a formal CNC course. Possibly you are enrolled in a technical school’s CNC course. Or maybe you are attending a company’s in-plant CNC training course. In either case, you have an instructor who is making presentations and available to help you understand the material. Or you may be using this text in conjunction with a video or CD-rom course, meaning your instructor is recorded. Either way, audio presentations should help you understand the concepts more easily. And with video and CD-roms, you can review the material as often as necessary.

If you have purchased this manual separately and do not have the benefit of an instructor, your task will be a little harder. While all of the key points made during live presentations are included in this text, you will be left on your own to study hard enough to grasp the material presented.

Scope

As the name of this text implies, we address but one of the three tasks a person must master in order to fully utilize a CNC machining center: programming.

Programming (which the focus of this text) is the act of preparing a series of commands that tell the CNC machining center how to machine a workpiece. It involves coming up with a machining process, selecting cutting tools, designing and documenting a setup, and actually creating the CNC program.

Setup is the act of preparing the CNC machining center to run a series of workpieces (called a *job* or *production run*). Tasks involve (among other things) making the workholding setup, assembling, measuring, and loading cutting tools, entering certain offsets, loading the program, and verifying that the program is correct. In this text, we will be discussing the related tasks in the approximate order that setups are made.

Operation actually involves two things. First, you must be comfortable with the general manipulation of a CNC machining center. This involves knowing the various components on the machine, its buttons and switches, and how to perform several important procedures. Second, and more importantly, you must be able to complete a production run once the setup is made. Tasks involve (among other things) workpiece load and unload, cycle activation, measuring completed workpieces and making sizing adjustments if necessary, and dull tool replacement.

Key Concepts approach

This effective presentation method will allow you to organize your thoughts as you read this text. This text includes six Key Concepts. Here are several benefits to this presentation method.

> 1) Any good training program should *put a light at the end of the tunnel*. All students want to know where they stand throughout any training course. With our Key Concepts approach, you will always have a clear understanding of your progress.

2) During each Key Concept, we will first present the main idea behind the concept. Again, it is at least as important to understand why you are doing things as it is to understand how to do them. Think of these early presentations for each Key Concept as the *why*. From there, we will present the specific techniques that are related to each concept.

3) The Key Concepts allow us to use a *building blocks approach* and present information in a very tutorial manner. We will be constantly building on previously presented information.

4) The Key Concepts approach allows us to limit the number of new ideas you must understand in order to grasp information presented within the text. Think of it this way: If you can understand but six basic ideas, you will be well on your way to becoming a proficient CNC programmer!

Lesson structure

We divide the six Key Concepts into nineteen lessons. This makes it possible to further organize the most important topics related to machining center usage.

Practice makes perfect

In this edition, we've included lots of practice and exercises within the text to help you confirm your understanding of the presented material. Answers are provided right in the text so you can quickly check your own work. This practice – and the ability to check your answers – changes this text from a book into a self-study course.

If you are using this text in conjunction with a technical school's CNC courses, you may be asked to submit your exercises without checking your answers. We ask that you do so honestly.

Key Concepts and lessons

Here is a list of the six Key Concepts and the sixteen lessons that comprise this text.

Key Concept 1: Know your machine from a programmer's viewpoint

1: Machine configurations
2: Visualizing program execution
3: Program zero and the rectangular coordinate system
4: Introduction to programming words

Key Concept 2: You must prepare to write programs

5: Preparation steps for programming

Key Concept 3: Understand the motion types

6: Programming the three most basic motion types

Key Concept 4: Know the compensation types

7: Introduction to compensation
8: Tool length compensation
9: Cutter radius compensation
10: Fixture offsets

Key Concept 5: You must provide structure to your CNC programs

11: Introduction to program structure
12: Four types of program format

Key Concept 6: Special features that help with programming

13: Hole-machining canned cycles
14: Working with subprograms
15: Other special programming features
16: Programming rotary devices

Enjoy!

We at CNC Concepts, Inc. wish you the best of luck with this text. We hope you find it easy to understand our written presentations and the presentations of your instructor (live or recorded). Once completed, we hope this text makes your introduction to CNC machining centers as easy and enjoyable as possible.

Lesson 1:

Machine Configurations

As a programmer, you must understand what makes up a CNC machining center. You must be able to identify its basic components – you must understand the moving components of the machine (called axes) – and you must know the various functions of your machine that are programmable.

Most beginners tend to be a little intimidated when they see a machining center in operation for the first time. Admittedly, there will be a number of new functions to learn. The first point to make is that you must not let the machine intimidate you. As you go along in this text, you will find that a machining center is very logical and is almost easy to understand with proper instruction.

You can think of any CNC machine as being nothing more than the standard type of equipment it is replacing with very sophisticated and automatic motion control added. Instead of activating things manually by hand-wheels and manual labor, you will be preparing a program that tells the machine what to do. Virtually anything that needs to be done on a true machining center can be activated through a program – meaning anything you need the machine to do can be commanded in a program.

There are two basic types of machining centers that we will be addressing in this text. They are vertical machining centers and horizontal machining centers. Let's start by describing the most common features of each.

Special note: Much of the material in this lesson is also included in the *CNC Machining Center Setup and Operation* self-study manual. If you have already read the setup and operation manual, please consider this lesson as a review.

Vertical machining centers

A vertical machining center has its spindle oriented in the vertical position. The cutting tool will be pointing downward toward the machine's table. This is a very popular type of CNC machining center because it closely resembles a popular type of conventional machine tool – the knee mill. For anyone having experience with a knee mill, a vertical machining center will appear quite familiar. Note that during machining, chips will tend to collect and build up on the workpiece, and may eventually interfere with machining operations.

Lesson 2

Visualizing The Execution Of A CNC Program

A CNC programmer must possess the ability to visualize movements a CNC machine will make as it executes a program. The better a person can visualize what the machining center will be doing, the easier it will be to prepare a workable CNC program.

Once again, we stress the importance of basic machining practice as it applies to CNC machining center usage. A machinist that has experience running a conventional milling machine will have seen machining operations taking place many times. While this experience by itself does not guarantee the ability to *visualize* a machining operation (seeing it happen in your mind), it dramatically simplifies the task of learning how to visualize a CNC program's execution.

When a machinist prepares to machine a workpiece on a conventional milling machine, they will have all related components needed for the job right in front of them. The machine, cutting tools, workholding setup, and print are ready for immediate use. It is highly unlikely that the machinist will make a basic mistake like forgetting to start the spindle before trying to machine the workpiece.

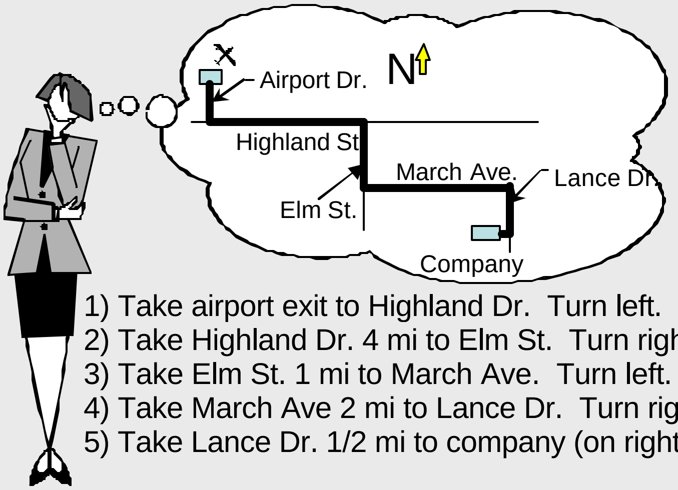
On the other hand, a CNC machining center programmer will be writing the program with only the workpiece drawing to reference. No tooling – no machine – and no workholding setup will be available to them. For this reason, a programmer must be able to visualize just exactly what will happen during the execution of the program - and this can sometimes be difficult, since this visualization must take place in the programmer's mind. A beginning programmer will be prone to forget certain things - sometimes very basic things (like turning the spindle on prior to machining the workpiece).

In this lesson, we will acquaint you with those things a programmer must be able to visualize. We will also show the first (elementary) program example to stress the importance of visualization.

Visualization is necessary to develop a set of instructions

Consider the visualization it takes to write a set of travel instructions to get someone from the airport to your company. Before you can write the instructions, you must be able to visualize the path from the airport to your company. If you cannot visualize the path, you can't write the instructions. Worse, if you think you can visualize the path (but you're wrong), you'll write incorrect instructions and the person following your instructions will get lost!

In similar fashion, if you cannot visualize the path a cutter will take as it machines the workpiece, you cannot write the CNC commands that drive the cutter through this path.



1) Take airport exit to Highland Dr. Turn left.
2) Take Highland Dr. 4 mi to Elm St. Turn right.
3) Take Elm St. 1 mi to March Ave. Turn left.
4) Take March Ave 2 mi to Lance Dr. Turn right.
5) Take Lance Dr. 1/2 mi to company (on right).

Lesson 3

Program Zero And The Rectangular Coordinate System

A programmer must be able to specify positions through which cutting tools will move as they machine a workpiece. The easiest way to do this is to specify each position relative to a common origin point called program zero.

You know that machining centers have three linear axes – X, Y, and Z. You also know these axes move and that they have a polarity (plus versus minus). In order to machine a workpiece in the desired manner, each axis must, of course, be moved in a controlled manner. One of the ways you must be able to control each axis is with precise *positioning control*.

In the early days of NC (even before CNC, over forty years ago), a programmer was required to specify drive motor rotation in order to cause axis motion. This meant they had to know how many rotations of an axis drive motor equated to the desired amount of linear motion for the moving component (table or headstock, for example). As you can imagine, this was extremely difficult – it was not logical. There is no relationship between drive motor rotation and motion of the moving component. Today, thanks to program zero and the rectangular coordinate system, specifying positions through which cutting tools will move is much easier.

The rectangular coordinate system has an origin point that we'll be calling *program zero*. It allows you to specify all positions (we'll be calling *coordinates*) from this central location. As a programmer, *you* will be choosing the location for program zero – and if you choose it wisely, many of the coordinates you will use in the program will come directly from your workpiece drawing, meaning the number and difficulty of calculations required for your program can be reduced.

Graph analogy

We use a simple graph to help you understand the rectangular coordinate system as it applies to CNC. Since everyone has had to interpret a graph at one time or another, we should be able to easily relate what you already know to CNC coordinates. Figure 1.13 is a graph showing a company's productivity for last year.

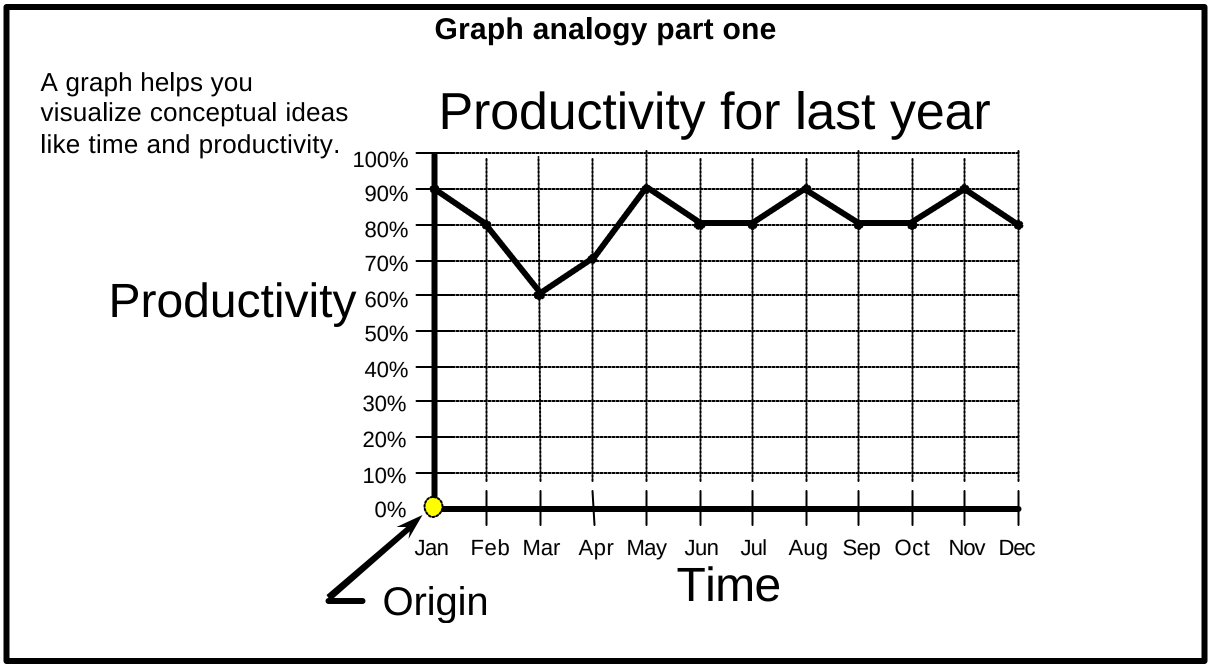


Figure 1.13 Graph example used to illustrate rectangular coordinate system

Lesson 4

Introduction To Programming Words

All CNC words include a letter address and a numerical value. The letter address identifies the word type. The numerical value (number) specifies the value of the word. You should be able to quickly recognize the most common ones.

You know from Lesson two that CNC programs are made up of commands – and that commands are made up of words. Words are categorized into types – and each word type has a special meaning to the machine. Each word type is designated by a *letter address*. You already know a few the letter addresses, like N for sequence number, G for preparatory function, X, Y, and Z for axis designations, S for spindle speed, F for feedrate, and M for miscellaneous (or machine) functions. In this lesson, we're going to introduce the rest of the word types.

If you are a beginner looking at the word types for the first time, you may want to read this section a few times to get better acquainted with these word types. Note that we are *not* asking you to memorize the word types – just to get familiar with them. In Key Concept number five – program formatting – we will provide you with a way to remember each word's function.

Also, this lesson is only intended to *introduce* each word, not to give you an in-depth description. When appropriate, we'll point you to the lessons that discuss the word type in more detail.

You will find that certain words are seldom used, meaning you will have little or no need for them. Other words are constantly used, and you will soon have them memorized after writing a few programs.

Some CNC words have more than one function, depending on commanded format. We will be showing you the primary (most common) function of the word next to the "A" description and the secondary use for the word next to the "B" description.

Once you've seen a word type a few times, it should not be too difficult to remember its function. Again, most word types are aptly named with a logical letter address. Additionally, only about fifty words are used consistently when programming, so try to look at learning to program a CNC machining center as like learning a foreign language that contains only fifty words.

As you continue with this text, *use this lesson as a reference*. If you come across a word or word type you don't recognize, remember to come back to this lesson. You've probably already noticed that we provide a quick reference for CNC words on the inside front and back cover of this text, but information in this lesson is a little more detailed.

Words allowing a decimal point

Current CNC controls allow you to include a decimal point in those words that are used to specify *real numbers* (values that require a portion of a whole number). You must remember to include a decimal point with these words or the control will revert to the fixed format for the word (as discussed in lesson two). Word types that allow a decimal point include:

A, B, C, X, Y, Z, I, J, K, F, Q, and R

Certain CNC words are used to specify *integers* (whole numbers). These word types do *not* allow a decimal point:

O, N, G, P, L, S, T, M, D, and H

Lesson 5

Preparation Steps For Programming

Any complex project can be simplified by breaking it down into small pieces. This can make seemingly insurmountable tasks much easier to handle. CNC machining center programming is no exception. Learning how to break up this complex task will be the primary focus of Lesson five.

As you now know, preparation will make programming easier, safer, and less error-prone. Now let's look at some specific steps you can perform to prepare to write CNC programs.

Prepare the machining process

Process sheets, also called routing sheets, are used by most manufacturing companies to specify the sequence of machining operations that must be performed on a workpiece during the manufacturing process. The person who actually prepares the process sheet must, of course, have a good understanding of machining practice, and must be well acquainted with the various machine tools the company owns. This person determines the best way to produce the workpiece in the most efficient and inexpensive possible way, given the company's available resources.

In most manufacturing companies, this involves *routing* the workpiece through a series of different machine tools and processes. Each machine tool along the way will perform only those operations the process planner intends, as specified on the routing sheet. This commonly means that non-CNC machine tools are needed to complete a given workpiece. For example, the square workpiece shown in practice exercise at the end of Lesson three might have the following routing sheet.

Op. #:	Operation:	Machine:
10	Procure material	Vender ID #12322
20	Cut bar stock to 3.2 long	Cut off saw
30	Clean and de-burr	Cleaning tanks
40	Mill contour, drill (9) holes	CNC machining center
50	Clean and de-burr	Cleaning tanks
60	Plate with nickel	Finishing tanks

Note that only operation 40 of this process requires a CNC machine tool. When a CNC machine is involved in the process plan, often the CNC machine will be required to perform several machining operations on the workpiece (as is the case in our example). As you know, all true CNC machining centers are multi-tool machines, meaning several tools can be used during one program. In some companies, the process sheet will clearly specify the order of machining operations that must be performed by the CNC machine. However, the vast majority of companies do not get so specific with their routing sheets. Instead, the sequence of machining operations to be performed on the CNC machine is left completely to the CNC programmer. If the programmer must develop the machining order, they must possess a good knowledge of basic machining practice.

In any event, the step-by-step machining order required to machine a workpiece on the CNC machine must be developed before the CNC program can be written. With a simple process, an experienced programmer may elect to develop the process as the program is being written. While some experienced programmers have the ability to do this, beginning programmers will find it necessary to plan the machining process first.

Lesson 6

Programming The Three Most Basic Motion Types

There are only three motion types used in CNC machining center programs on a regular basis – rapid, straight-line, and circular motion. You must understand how they are commanded.

While it helps to understand how the machining center will interpolate motion, it is not as important as *knowing how to specify motion commands in a program*. This is the focus of Lesson six. Let's begin our discussion by showing those things that all motion types share in common.

Motion commonalties

All motion types share five things in common:

First, they are all modal, meaning a motion type will remain in effect until it is changed. If more than one consecutive movement of the same type must be made, you need only include the motion type G code in the *first* command of the series of movements.

Second, each motion type command requires the *end point* of the motion. The control will assume the tool is positioned at the starting point of the motion prior to the motion command. Think of motion commands that form a tool path as being like a series of connect-the-dots.

Third, all motion commands are affected by whether or not you specify coordinates in the absolute or incremental positioning mode. In the absolute positioning mode (specified by G90), specified end points will be relative to the program zero point. In the incremental positioning mode (specified by G91), specified end points will be relative to the tool's current position. As stated in Lesson three, you should concentrate on specifying coordinates in the absolute positioning mode.

Fourth, each motion command requires *only* the moving axes. If specifying a motion in only one axis, only one axis specification (X, Y, or Z) need be included in the motion command. *Axes that are not moving can be (and should be) left out of the command.*

Fifth, leading zeros can be left out of the G codes related to motion types. This means the actual G codes used to instate the motion types can be programmed in one of two ways. G00 and G0 (stated G zero-zero and G zero) mean exactly the same thing to the control, as do G01 and G1, G02 and G2, G03 and G3. All examples in this text do include the leading zero.

Understanding the programmed point of each cutting tool

In order to generate correct motions with your cutting tools, you must understand the position on the tool that you are programming. In many cases, you will need to calculate programmed coordinates – not just from the blueprint – but based on some cutting tool criteria as well. Figure 3.4 shows a series of center-cutting tools used to perform hole-machining operations.

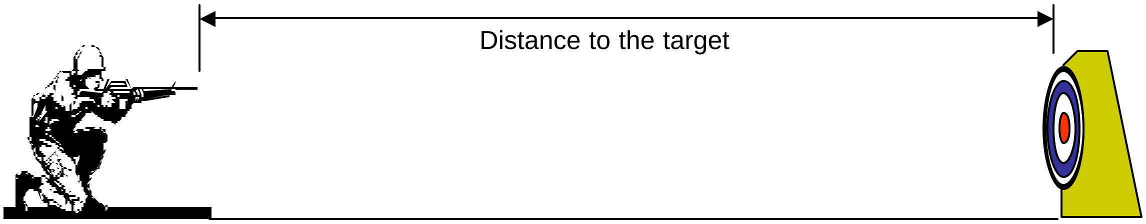
Lesson 7

Introduction To Compensation

An airplane pilot must compensate for wind direction and velocity when setting a heading. A race-car driver must compensate for track conditions as they negotiate a turn. A marksman must compensate for the distance to the target when firing a rifle. And a CNC programmer must compensate for certain tooling-related problems as programs are written.

What is compensation and why is it needed?

When you compensate for something, you are allowing for some unpredictable (or nearly unpredictable) variation. A *race car driver* must compensate for the condition of the race track before a curve can be negotiated. In this case, the unpredictable variation is the condition of the track. An *airplane pilot* must compensate for the wind direction and velocity before a heading can be set. For them, wind direction and velocity are the unpredictable variations. A *marksman* must compensate for the distance to the target before a shot can be fired – and the distance to the target is the unpredictable variation. The marksman analogy is remarkably similar to what happens with most forms of CNC compensation. Let's take it further…



Before a marksman can fire a rifle, they must judge the distance to the target. If the target is judged to be fifty yards away, the sight on the rifle will be adjusted accordingly. When the marksman adjusts the sight, they are *compensating for the distance to the target.* But even after this preliminary adjustment and before the first shot is fired, the marksman cannot be *absolutely* sure that the sight is adjusted perfectly. If they've incorrectly judged the distance – or if some other variation (like wind) affects the sight adjustment – the first shot will not be perfectly in the center of the target.

After the first shot is fired, the marksman will know more. If the shot is not perfectly centered, another adjustment will be needed. And the second shot will be closer to the center of the target than the first. Depending upon the skill of the marksman, it might be necessary to repeat this process until the sight is perfectly adjusted.

With *all* forms of CNC compensation, the setup person will do their best to determine the compensation values needed to perfectly machine the workpiece (just as the marksman does their best in judging the distance to the target and adjusting the sight). But until machining actually occurs, the setup person cannot be sure that their initial compensation values are correct. After machining, they may find that another variation (like tool pressure) is causing the initial adjustment to be incorrect. Depending upon the tolerances for the surfaces being machined, a second adjustment may be required. After this second adjustment, machining will be more precise.

There is even a way to make an initial adjustment (prior to machining) that ensures excess material will remain on the machined surface after the first machining attempt (this technique is called *trial machining*). This guarantees that the workpiece will not be scrapped when the cutting tool machines for the first time – and is especially important for very tight (small) tolerances. With tight tolerances, even a small machining imperfection will cause a scrap workpiece.

Lesson 8

Tool Length Compensation

Tool length compensation allows a programmer to ignore the precise length of each tool as a program is written. It is used for every tool in every program you write – so you must understand this important CNC feature.

You know that program zero assignment values for the X and Y axes are entered into fixture offsets from the *spindle center* to the program zero point in X and Y (while the machine is at its zero return position). So when you specify a position of X1.0 Y1.0 in a program, the machine will be able to send the *spindle center* (and tool center) to this position – relative to program zero.

In the Z axis, you know that the program zero assignment value is entered into the fixture offset from the *spindle nose* to the Z axis program zero surface (again, while the machine is at its zero return position). But for cutting tool positioning, you don't want to specify Z axis positions from the *spindle nose*. This would be very cumbersome – and it would require that you know the precise length of each tool before you could even write the program. Instead, you specify Z axis positions to the *tip* of each cutting tool. That is, when you specify a position of Z0.1, the *tool tip* will move to this position. In order to be able to program the tool tip in the Z axis, a feature called *tool length compensation* must be used. Mastering tool length compensation is the focus of lesson number eight.

The reasons why tool length compensation is needed

Cutting tools used on machining centers differ from one another. For one thing, there are a variety of cutting tool *types* that are used on machining centers, including center drills, spot drills, drills, taps, reamers, boring bars, end mills, and face mills (among many others). Each type of tool requires a different way of gripping the actual cutting tool in its holder. Some tools (like some straight shank tools) use a collet system. Others (like end mills) use a set-screw to hold the cutting tool in place. Yet others (like face mills and taps) require a very special style of tool holder – designed especially for the cutting tool.

No two tools will have exactly the same length

Given the vast assortment of cutting tools available for use on CNC machining centers, it is unlikely that any two tools used in a program will be exactly the same length. And you will not know precisely how long each tool will be when you write the program. Figure 4.4 shows five different types of cutting tools to illustrate this point.

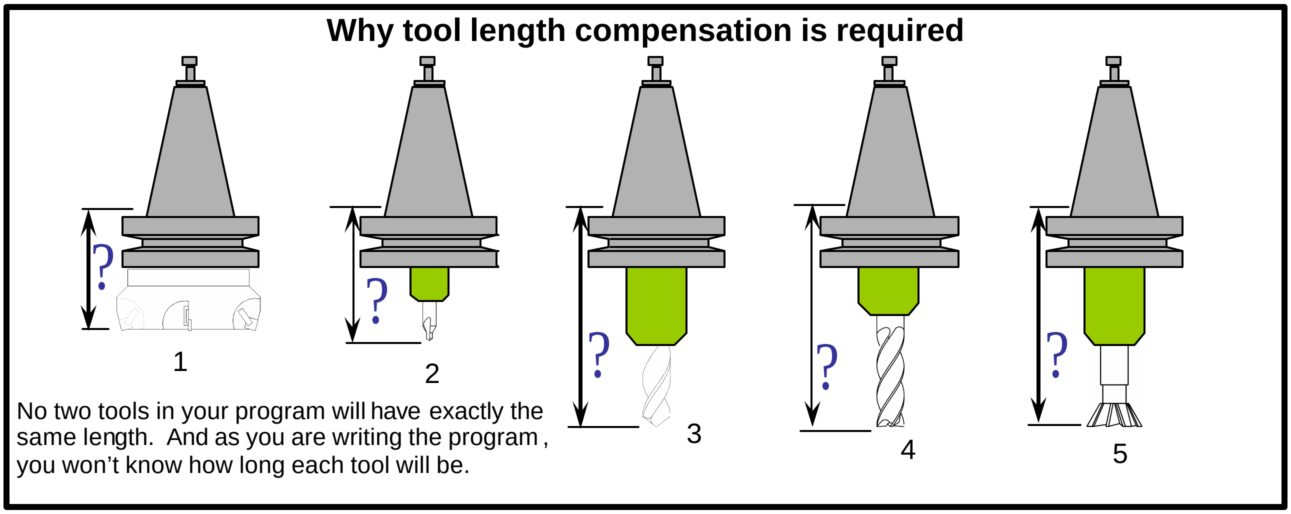

Why tool length compensation is required
?
?
?
?
?
1
2
3
4
5
No two tools in your program will have exactly the same length. And as you are writing the program , you won't know how long each tool will be.

Figure 4.4 – Five cutting tools that might be used in by a CNC machining center program

Lesson 9

Cutter Radius Compensation

Cutter radius compensation is only used for milling cutters. Just as tool length compensation lets you ignore the exact length of cutting tools as you write programs, cutter radius compensation allows you to ignore the precise diameter of milling cutters used for contour milling.

You know from Lesson six that milling cutters (like end mills) can be used for contour milling operations. Motions can be linear (straight-line) or circular. To this point, we have shown the contour milling tool path based upon a milling cutter's *centerline* path. As you know, calculating coordinates for a milling cutter's centerline path requires that you consider the milling cutter's radius for *every* coordinate – which can make calculating coordinates quite difficult.

In Lesson nine, we're going to show you how to minimize the calculations that must be done when determining coordinates needed for contour milling. Instead of using the milling cutter's centerline coordinates, you will be using coordinates that are right on the work surface to be milled. And again, these coordinates are much easier to calculate. Mastering the programming of cutter radius compensation will be the focus of Lesson nine.

Will you need to learn this feature?

Unlike tool length compensation – which is used for every cutting tool in every program – cutter radius compensation is only used for milling cutters, and only when side milling (milling on the periphery – outside diameter – of the cutter). If your company doesn't perform any side milling operations, you won't be needing cutter radius compensation.

Even if this is the case, you will still want to know the reasons *why* cutter radius compensation is required for contour milling. So at the very least, read on until we start discussing how cutter radius compensation is programmed. You may then want to skip to Lesson ten. If the need ever arises, you can always come back to this lesson and dig in.

Reasons why cutter radius compensation is required

Let's begin by discussing why you must master cutter radius compensation. Some of the reasons are quite similar to the ones given for tool length compensation.

Calculations are simplified for manual programmers

When performing contour milling operations *without* cutter radius compensation (as shown in Lesson six), you must specify coordinates for the milling cutter's centerline path. This requires that you consider the size of the milling cutter in *every* calculation – complicating each calculation. With cutter radius compensation, and as a manual programmer (not using a computer aided manufacturing [CAM] system), you will specify coordinates that are right on the work surface, ignoring the size of the milling cutter. Figure 4.8 shows the two different tool paths.

Lesson 10

Fixture Offsets

Tool length and cutter radius compensation allow you to ignore certain attributes of cutting tools as you writer your CNC programs. In like fashion, fixture offsets allow you to ignore the precise location of the work holding device/s on the machine table as you write programs.

You know from Lesson three that fixture offsets are used to assign program zero. Based upon our discussions so far, you know that the distances from the zero return position to the program zero point must be determined (either measured or calculated). The polarity of these values is almost always negative since the zero return position is usually close to the plus over-travel limit of each axis. These values are entered into fixture offset number one to assign program zero. Again, the specific techniques used to assigned program zero are shown in the self-study manual: *CNC Machining Center Setup and Operation* – sold separately.

We've only shown how to use *one* program zero point (requiring the use of fixture offset number one). In this lesson, we'll be showing when *multiple* program zero points are required as well as how they are assigned and programmed.

> Fixture offsets should be your program zero assignment method-of-choice over using G92 in the program. ***You should only use G92 to assign program zero if your machining center does not have fixture offsets.***

Do you need to learn about fixture offsets?

There are many companies that don't need any more from fixture offsets than we show Lesson three and in the *CNC Machining Center Setup and Operation* manual. They never have more than one program zero point per program. The setup person measures program zero assignment values during setup and enters them into fixture offset number one. They're satisfied using fixture offsets in this manner – and have no need for anything more.

However, when it comes to becoming more efficient with program zero assignment (and in turn – reducing setup time), many companies can benefit by improving their methods. You cannot become more efficient, of course, unless you know something better is possible.

While the material we present in this lesson may not be of immediate need to you, you will definitely want to understand its content. If you do decide to skip ahead, be sure to come back to this material once you've acquainted yourself with your machining center and its usage.

Lesson 11

Introduction To Program Structure

Structuring a CNC program is the act of writing a program in a way that the CNC machine can recognize and execute safely, efficiently, and with a high degree of operator-friendliness.

You know that CNC programs are made up of commands, that each command is made up of words, and that each word is made up of a letter address and a numerical value.

You also know that programs are executed sequentially – command by command. The machine will read, interpret, and execute the first command in the program. Then it will move on to the next command. Read – interpret – execute. It will continue doing so until the entire program has been executed.

And you have seen several complete programs – you have even worked on a few if you have done the exercises in this text and in the workbook. You have probably noticed that there is quite bit of consistency and *structure* in CNC programs.

Our focus in Lesson eleven will be to help you understand more about the structure that is used in CNC programming.

Objectives of your chosen program structure

CNC machines have come a long way. In the early days of NC (before computers), a program had to be written *just so*. If *anything* was out-of-place, the machine will go into an alarm state – failing to execute the program. While today's CNC machines are *much* more forgiving, you must still write CNC programs in a rather strict manner.

There are many ways to write a workable program – and the methods you use in structuring your programs will have an impact on three important objectives:

✍Safety
✍Efficiency
✍Ease-of-use (operator friendliness)

It may be impossible to come up with a perfect balance among these objectives. Generally speaking, what you do to improve one objective will negatively affect the other two. When faced with a choice, a beginning programmer's priorities should always lean toward *safety* and *ease-of-use*. Our recommended programming structure stresses these two objectives. We will, however, show some of the efficiency-related short-comings of our recommended methods – so you can improve efficiency as you gain proficiency.

We're going to be assuming that *you have control* of the structure you use to write programs. Your company may, however, already have a programmer that is writing programs with a different structure. As long as these programs are working – and satisfying the company's objectives – you're going to have to adapt to the established structure. If you understand the reasons for formatting, and if you understand one successful method for structuring programs, it shouldn't be too difficult to adapt.

Reasons for structuring programs with a strict and consistent format

Let's begin by discussing the reasons *why* you must write your programs using a strict structure.

Familiarization

You must have some way to get familiar with CNC programming. You'll need some help writing your first few programs. The formats we show in Lesson twelve will provide you with this help. You'll be able to use our given formats as a crutch until you (eventually) have them memorized.

Lesson 12

Four Types Of Program Format

The formats shown in this lesson will keep you from having to memorize most of the words and commands needed in CNC programming. As you'll see, a large percentage of most programs is related to structure.

You know the reasons why programs must be formatted using a strict structure. In Lesson twelve, we're going to show the actual formats. We will show two sets of format, one for vertical machining centers and another for horizontal machining centers. We will also explain every word in each format in detail. We will also show an example program that stresses the format's use.

There are four types of program format:

Program startup format

Tool ending format

Tool startup format

Program ending format

Any time you begin writing a new program, follow the program start-up format. You can copy this *structure* to begin your program. The actual values of some words will change based on what you wish to do in your own program, but the structure will remain the same every time you begin writing a new program.

After writing the program start-up format, you write the motions for the first tool's machining operations. When finished with the first tool motions, you follow the tool ending format. You then follow the tool start-up format for the second tool and write the second tool's machining motion commands. From this point, you toggle among tool ending format, tool startup format, and machining motion commands until you are finished machining with the last tool. You then follow the program ending format.

One of the most important benefits of using these formats is that you *will not have to memorize anything*. You simply copy the structure of the format.

Our formats assume that you are using fixture offsets to assign program zero. Later in this lesson, we'll discuss the changes that must be made to these formats if you must use G92 in the program to assign program zero. (Again, you should only use G92 to assign program zero if your machining center does not have fixture offsets.)

Format for vertical machining centers

This format is used for vertical machining centers that have fixture offsets. This format assumes that the machine has a double-arm automatic tool changer, that the tool change position is the Z axis zero return position, and that the machine is resting at the tool change position in Z when the program begins. When this program ends, the machine is left at the Y and Z axis zero return position. This makes a convenient workpiece loading position (with the table out toward the operator in the Y axis).

Again, you are to use the strict *structure* of these given formats. But the values that are shown in **bold** will change from program to program and from tool to tool.

Lesson 13

Hole-Machining Canned Cycles

Armed only with what you know so far, programming hole-machining operations is very tedious. Hole-machining canned cycles will dramatically simplify the programming of these very common machining operations.

Almost all programs have at least *some* hole-machining operations. If you have been doing the exercises in this text and in the workbook, you have seen how tedious, time consuming, and error-prone it can be to program hole-machining operations with G00 and G01. You know that with G00 and G01, each hole will require *at least* three commands, making your program quite long. And we've only performed basic drilling operations. Peck drilling, tapping, boring, and counter-boring operations will require even more commands per hole.

Canned cycles will dramatically simplify the programming of hole-machining operations. *Only one command is required per hole*, regardless of the machining style (drill, peck drill, tap, ream, bore, counter-bore, etc.). Additionally, canned cycles are *modal*, meaning once you instate a canned cycle, you can continue machining holes by simply listing hole-positions. Again, this will dramatically shorten the program's length, make programming easier, less time-consuming, and less error-prone.

The meaning of "canned"

A *canned* cycle is a series of preset movements that the CNC machine will execute based upon a limited amount of program information. The *zero return command* (G28) is a simple kind of canned cycle. G28 actually makes the machine do *two* things. First, the machine will move to the *intermediate position*. Second, it will move to the *zero return position*.

If you have the *single block* switch turned on (a function that makes the machine execute one command in the program at a time), you actually have to activate the cycle twice to make the control complete the G28 command. Again, the first time you activate the cycle (by pressing the *cycle start* button), the machine executes the motion to the intermediate position. The second time, it moves to the zero return position.

Hole-machining canned cycles are much more elaborate. Even the standard drilling cycle will make the machine do at least three things per command. With the chip-breaking peck drilling cycle, one command can actually generate over one-hundred movements.

Here's how simple it is to use canned cycles: You *instate* the canned cycle for the first hole to machine. With a standard drilling cycle, for example, the instating command includes the cycle type (G81), the first hole position in XY, the hole's bottom position in Z (commonly its depth), and the machining feed rate. The instating command actually machines the first hole. To machine the rest of the holes, you simply list their XY coordinates, one set per command. After the last hole, you must *cancel* the cycle with a G80 word.

Here is a list of the most common hole-machining canned cycles in approximate order of popularity:

G80 – Cancel any of the canned cycles
G81 – Standard drilling cycle
G73 – Chip-breaking peck drilling cycle
G83 – Deep hole peck drilling cycle
G82 – Counter-boring cycle
G84 – Right hand tapping cycle
G74 – Left hand tapping cycle
G86 – Standard boring cycle

Lesson 14

Working With Subprograms

There are times when a series of CNC commands must be repeated – within one program – and sometimes among several programs. Whenever you find yourself writing a series of commands a second time – you should consider using a subprogram. The longer the series of commands and the more often they must be repeated, the more a subprogram can help.

You know that a CNC machining center will execute a program in sequential order. It will start with the first command in the program: read it – interpret it – and execute it. Then it will move on to the next command – read, interpret, execute. It will continue this process for the entire program.

You also know that there are times when a series of commands in your program must be repeated. We've shown two times so far. One time is when using cutter radius compensation to rough and finish mill using the same set of coordinates. You must program the finishing tool path coordinates twice, once for the rough milling cutter (using an offset value that is larger than the actual cutter size) and once for the finish milling cutter.

Another time when commands must be repeated is when machining holes that require multiple machining operations (like center drilling, drilling, and tapping a series of holes) – the more holes that must be machined, the more commands that must be repeated.

In Lesson fourteen, you will learn about a way to change the order of program execution to some extent – and this will be especially helpful when commands must be repeated.

The difference between main- and sub- programs

A *main program* is the program that a setup person or operator will execute when they activate a cycle. *Every program shown to this point in the text has been a main program.* Main programs almost always end with M30 (or M02 with some machines).

A *subprogram* is a program that is invoked by a main program or by another subprogram. A subprogram ends with M99.

In a main program, when you reach a series of commands that you know will be repeated, you can invoke a subprogram that contains the repeated commands (you place the repeated commands in the subprogram instead of in the main program). Each time the commands must be repeated, you will invoke the same subprogram.

M98 is the word used to invoke a subprogram. A *P word* in the M98 command specifies the program number of the subprogram to be executed. When the machine is finished executing the subprogram, it will come back to the main program to the command after the calling M98.

Figures 6.11 and 6.12 show a simple application for a subprogram.

Lesson 15

Other Special Programming Features

Current model CNC machining centers come with many features to help with special applications. While some of these features will be of little need to you in the immediate future, it is important to know they exist. You cannot begin to apply any feature that you don't know about.

CNC control manufacturers strive to equip their controls with as many helpful programming features as possible. Those mentioned so far (canned cycles and subprograms) are used on a very regular basis — and you should strive to master them. However, there are some special programming features that are not used nearly as regularly. Indeed there are some features that are extremely important to one company but of no value to another.

As you study this lesson, you need to consider your own company's CNC applications. If you are in doubt about the value of a given feature, ask your instructor or an experienced person in your company about its value to your company. You can minimize your studies about those features your company does not currently need. You can always come back and study this lesson in greater detail should the need arise.

As you study this lesson, remember that *your ingenuity is based predominantly upon your knowledge of what is possible*. You cannot apply a feature of which you are unaware. At the very least, this lesson will acquaint you with what is possible with special CNC programming features.

The organization of this lesson is not as tutorial as previous lessons. While we will explain each feature in detail and in tutorial format, we don't present them in a special order. Here are the topics contained in this lesson:

✍Block delete (also called optional block skip)

✍Special techniques with sequence numbers

✍G codes that have not yet been introduced (in numerical order)

Block delete (also called optional block skip)

The block delete function is used to give the CNC operator a choice between one of two possibilities. An on/off switch on the control panel (commonly labeled block delete or optional block skip) is used to actually make the choice. Since applications for block delete vary, the programmer must make each use of block delete *very clear* to the operator. This should be done in the setup- and/or production-run-documentation.

A slash code (/) in the program tells the control to look to the position of the block delete switch. If the switch is on, the control will skip any words to the right of the slash code. If the switch is off, the control will execute these words.

Here is a simple example. Say your machining center does not have an adequate coolant switch. Your setup person has no way to turn off the coolant when programs are being verified (whenever an M08 is executed, coolant will come on). To solve this problem, you can place a slash code at the beginning of every coolant command. Here is one way to do so.

```
/ N015 M08 (If the block delete switch is on, the M08 will be skipped and coolant will stay off)
```

During program verification, the setup person can turn *on* the block delete switch. This will force the control to *skip* commands that turn coolant on (leaving the coolant off).

The slash code does not have to be placed at the beginning of a command. If for example, you have a totally enclosed work area, you may want the coolant to come on during each cutting tool's approach to the workpiece. Consider this command.

Lesson 16

Programming Rotary Devices

Many CNC machining centers, and especially horizontal machining centers, are equipped with a rotary device that allow more than one workpiece surface to be exposed to the spindle for machining during the CNC cycle. This lesson will explain how they are programmed.

This lesson will only apply to you if your machining center has some kind of rotary device that allows the workpiece to be rotated during the CNC cycle. For *vertical* machining centers, you may have an indexer or rotary table (with full rotary axis) that rests *on top of the table.* For horizontal machining centers, your machine is likely to have an indexer or rotary axis built into the table of the machine.

The difference between an indexer and a rotary axis

The major difference between an indexer and a rotary axis has to do with whether machining can occur *during rotation*. With an indexer, it cannot. Only a true rotary axis will allow machining during rotation. Frankly speaking, the vast majority of applications for rotary devices do not require machining during rotation. Instead, the rotary device is simply used to expose different surfaces of the workpiece to the spindle for machining. Machining is done *after rotation* – which either rotary device can do. *A rotary axis can be used as an indexer but an indexer cannot be used as a rotary axis.*

A note to horizontal machining center programmers

All example programs to this point have been in the format for a vertical machining center. And we have tried to keep our examples as easy to understand as possible. Though our emphasis has been for vertical machining centers, note that every presentation to this point will also apply to horizontal machining centers – and – if you do have a horizontal machining center, it is mandatory that you understand the points made thus far.

The major usage difference between vertical and horizontal machining centers is that most horizontal machining centers are equipped with some form of rotary device within the machine's table. And by the way, if a horizontal machining center does not have a rotary device (some horizontal boring mills fit into this category), it is programmed in *exactly* the same manner as a vertical machining center.

Any of the programs shown in this text so far can be run on a horizontal machining center without major modifications. Only the minimal differences in program formatting shown in Lesson twelve need to be done (changing movements to the tool change position). When you think about it, a horizontal machining center is nothing more than a vertical machining center that has been placed on its back.

When a rotary device is used, either on a vertical or horizontal machining center, programs get longer. They don't necessarily become more difficult to write, they just get longer. Again, several surfaces of the workpiece can be machined in one program, so more tools can be used by a program. (And by the way, the automatic tool changer magazines on horizontal machining centers tend to have a much larger capacities than those used on vertical machining centers. Indeed, some can hold well over one-hundred tools.)

Additionally, most horizontal machining centers are equipped with *pallet changers*, meaning two or more jobs can be running on the machine at a given time (one workpiece is machined in the work area while the operator loads another). This means at least two programs will be needed in the machine at any given time.

The most difficult part of programming for a machine that has a rotary device is related to preparation and organization. Since more than one workpiece surface can be machined – and more cutting tools will be needed – the programmer must carefully plan the process. Developing a sequence of machining operations is

Quiz

1) Which style of machining center is the most popular?

A) C-frame style vertical C) Bridge-style vertical
B) Knee-style vertical D) Horizontal

2) Specify the correct axis for each moving component below (C-frame style vertical machining center).

a) ____Table motion in/out
b) ____Headstock motion up/down
c) ____Table motion left/right

3) Specify the correct axis and polarity for each tool motion below (C-frame style vertical machining center).

a) _____Tool motion left
b) _____Tool motion away from you
c) _____Tool motion down

4) Provide the CNC word or command needed to activate the following:

a) Start spindle (fwd) at 300 rpm: ____________
b) Start spindle (reverse) at 2,000 rpm: ____________
c) Stop spindle: ____________
d) Turn on the flood coolant: ____________
e) Turn off the coolant: _____________
f) Place tool station seven in spindle: ____________
g) Select inch mode: ____________

Talk with experienced people in your company to learn more: Do your machining centers have more than one spindle range? If so, what are the cut-off points for each range? At what rpm does the spindle achieve maximum horsepower in each range? Do any of your machines have high pressure coolant systems? If so, what is the related M code? How many cutting tools can your machining centers hold?

Answers: 1: A, 2a: Y, 2b: Z, 2c: X, 3a: X-, 3b: Y+, 3c: Z-, 4a: S300 M03, 4b: S2000 M04, 4c: M05, 4d: M08, 4e: M09, 4f: T07 M06, 4g: G20

STOP! Do practice exercise number one at the end of this manual.

Practice exercises and programming activities

The pages that follow contain the exercises you have been asked to do after each lesson. If you are using this text in conjunction with a technical school class, you may be asked by your instructor to tear out and turn in some of these exercises for grading.

Machine configurations 1

Name: Date: Score (100 possible):

10 points each

1) The single-most important topic a machining center programmer must understand is the basic machining practice of machining center operations.

✍ true
✍ false

2) The orientation of the spindle determines whether the machining center is a *vertical* or *horizontal* machining center.

✍ true
✍ false

3) When it comes to programming, unless the vertical machining center has a rotary device, horizontal machining center programs tend to be shorter than vertical machining center programs.

✍ true
✍ false

4) Name and describe the three linear directions of motion (linear axes) for a vertical machining center.

5) Name and describe the directions of motion (axes) for a horizontal machining center. Include the most common rotary axis.

6) From a programmer's viewpoint, describe the *plus* direction for each linear axis of a vertical machining center.

7) Name two ways that you can control of the spindle of a machining center (of three possible ways).

8) How is feedrate designated on most machining centers?

✍ a. in feed per revolution
✍ b. in rpm
✍ c. in surface feet per minute
✍ d. in feed per minute

9) What is the letter address used to specify spindle speed?

✍ a. M word
✍ b. F. word
✍ c. S word
✍ d. R word

10) What is the letter address used to specify feedrate?

✍ a. M word
✍ b. F. word
✍ c. S word
✍ d. R word

1 Tests comprehension of motion types

Name: Date: Score (126 possible):

Instructions: This is the first complete program you will be working on. Though it will not require that you actually write the entire program on your own, it will show you much of what is involved in doing so.

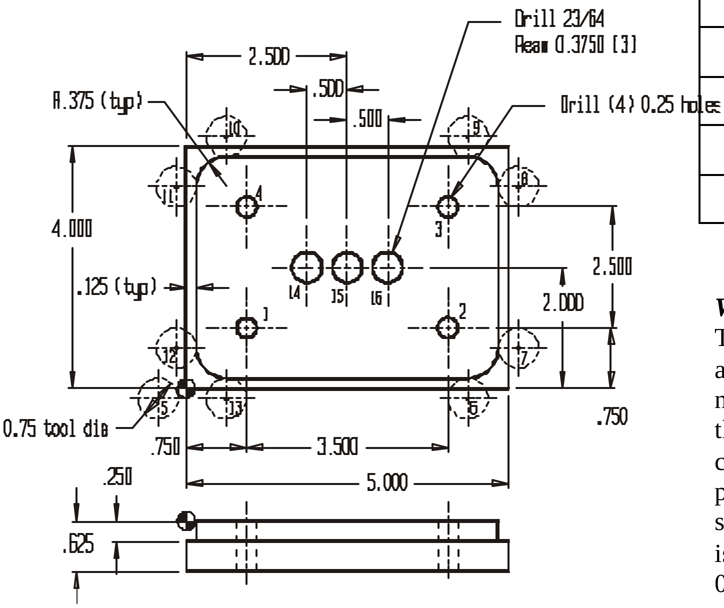
First use the print to come up with the coordinates going into the program by filling in the coordinate sheet. For the Z coordinate of the hole machining operations, you may wish to give two values, one for the approach position and the other for the hole bottom position, so you'll have these values available when you work on the program. For the 0.25 holes, for example, you could specify Z as 0.100/-0.730. For the 0.75 end mill, position it 0.100 away from the workpiece in X in point number 5.

Second, study the process to confirm that you understand what the program will be doing. *Third*, fill in the blanks in the program with the appropriate G codes, feedrates, spindle speeds, and axis positions.

Coordinate sheet:

#	X	Y	Z
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Print:



Warning about the drawing: The drawing is not to scale. The end mill is drawn smaller than its actual size. While it appears that the end mill is tangent to the center of hole number one (and 2, and 3, and 4), in reality it is not. Notice the 0.125 step around the workpiece. This must be used to calculate the positions for the end mill's path. For the X position of point 13, for example, you must add the 0.125 step to the 0.375 workpiece radius, the result for point 13's X is 0.5, ***not 0.375*** as it would be if you (incorrectly) subtract 0.375 (tool radius) from the 0.75 X position of point one.

Process:

Seq.	Operation description	Tool	Station	Speed	Feedrate
1	Center drill all holes	# 3 center drill	1	2000 rpm	5.0 ipm
2	Drill (4) 0.25 holes	1/4 drill	2	1200 rpm	4.8 ipm
3	Drill (3) 23/64 holes	23/64 drill	3	800 rpm	.0 ipm
4	Ream (3) 0.3750 holes	0.375 reamer	4	600 rpm	7.0 ipm
5	Mill contour	0.750 end mill	5	400 rpm	4.5 ipm

Program: (One point for each correctly filled in blank)

```
O0001 (Program number)

(Center Drill)
N005 T01 M06
N010 G54 G90 S____ M03 T02
N015 G___ X______ Y______ (pt 1)
N020 G43 H01 Z______ M08
N025 G01 Z-0.200 F5.0
N030 G00 Z______
N035 ______ (pt 2)
N040 ____ ______
N045 ____ ______
N050 ______ (pt 3)
N055 ____ ______
N060 ____ ______
N075 ______ (pt 4)
N080 ____ ______
N085 ____ ______
N090 ______ ______ (pt 14)
N095 ____ ______
N100 ____ ______
N105 ______ (pt 15)
N110 ____ ______
N115 ____ ______
N120 ______ (pt 16)
N125 ____ ______
N130 ____ ______ M09
N135 G91 G28 Z0 M19
N140 M01

(1/4 drill)
N145 T02 M06
N150 G54 G90 S____ M03 T03
N155 G___ X______ Y______ (pt 1)
N160 G43 H02 Z______ M08
N165 G01 Z-0.730 F_____
N170 ____ ______
N175 ______ (pt 2)
N180 ____ ______
N185 ____ ______
N190 ______ (pt 3)
N195 ____ ______
N200 ____ ______
N205 ______ (pt 4)
N210 ____ ______
N215 ____ ______ M09
N220 G91 G28 Z0 M19
N225 M01

(23/64 drill)
N230 T03 M06
N235 G54 G90 S____ M03 T04
N240 G___ X______ Y______ (pt 14)
N245 G43 H03 Z______ M08
N250 G01 Z______ F_____
N255 ____ ______
N260 ______ (pt 15)
N265 ____ ______
N270 ____ ______
N275 ______ (pt 16)
N280 ____ ______
N285 ____ ______ M09
N290 G91 G28 Z0 M19
N295 M01

(0.3750 reamer)
N300 T04 M06
N305 G54 G90 S____ M03 T05
N310 G___ X______ Y______ (pt 14)
N315 G43 H04 Z______ M08
N320 G01 Z______ _____
N325 ____ ______
N330 ______ (pt 15)
N335 ____ ______
N340 ____ ______
N345 ______ (pt 16)
N350 ____ ______
N355 ____ ______ M09
N360 G91 G28 Z0 M19
N365 M01

(0.750 end mill)
N370 T05 M06
N375 G54 G90 S____ M03 T01
N380 G___ X______ Y______ (pt 5)
N385 G43 H05 Z0.1 M08
N390 G01 Z-0.25 F50.
```

N395 ______ F_____(pt 6)
N400 ____ ______ ______ ______
(pt7)
N405 ____ ______ (pt 8)
N410 ____ ______ ______ _____
(9)
N415 ____ ______ (pt 10)

N420 ____ ______ _____ _____ (11)
N425 ____ ______ (pt 12)
N430 ____ ______ _____ _____ (13)
N435 G00 Z0.1 M09
N440 G91 G28 Z0 M19
N445 M30

Answers to Exercises

If you are attending a class that uses this text, your instructor may ask you not to look at these answers prior to submitting your work. (Each exercise can be removed from this text without removing any content.) On the other hand, if you are reading this text on your own, you can use these answers to check your work.

Exercise 1: Machine configurations

1) true
2) true
3) false
4) a. The X axis is the left/right direction (as viewed from the front of the machine)
   b. The Y axis is the in/out direction (as viewed from the front of the machine)
   c. The Z axis is the up/down direction (as viewed from the front of the machine)
   d. If the machine has a rotary table on the table, it could be called the A, B, or C axis, depending upon orientation
5) a. As viewed from the spindle, left/right motion is the X axis.
   b. As viewed from the spindle, up/down motion is the Y axis.
   c. As viewed from the spindle, in/out motion is the Z axis.
   d. If the machine has a rotary axis mounted in the table, it is called the B axis.
6) a. As viewed from the front, tool motion toward the right
   b. Tool motion away from the operator
   c. Tool motion in an upward direction
7) Answers include speed, direction and possibly range
8) d. in feed per minute
9) c. S word
10) b. F word

Extension question answers:

11) All roughing should be done before any finishing is done.
12) When the machine is rigid enough to withstand the tendency for the tool to pull into the cut. Most machinists agree that climb milling will render a better finish than conventional milling, but should not be done with knee style machines.
13) Because with most machines, the tool does not actually move along with one or more of the axes. To try to consider table motion plus versus minus can be quite confusing.

Exercise 2: Visualizing program execution

1) The person performing the operation manually has everything in front of them and available. The person preparing the CNC program must do so with only print, pen, and paper.
2) true
3) true
4) true
5) false
6) true
7) false
8) a. X0.0007
9) A modal word will set a state the remains in effect until changed or canceled.
10) An initialized word will be automatically instated when the machine power is turned on.

Extension question answers:

11) In order to write CNC programs, a programmer must be able to visualize machining operations as the program is