MITSUBISHI ELECTRIC
Changes for the Better

PLC
Efficient Programming

This course describes the use of structured programming to make programming more efficient. This course is aimed at those who have been programming using simple ladder diagrams.

L(NA)00205ENG

Introduction | **Purpose of the course**

Through this course, you will learn the advantages of structured programming. We hope this course will help you program more efficiently.

> The following course is a prerequisite prior to taking this course:
>
> - Programming Basic

The contents of this course are as follows.

Chapter 1 - Why structured programming is necessary?

This chapter describes the necessity of structured programming for better efficiency.

Chapter 2 - Why structured programs are so efficient

This chapter describes the benefits of structured programming.

Final Test

Pass grade: 60% or higher

Go to the next page	▶	Go to the next page.
Back to the previous page	◀	Back to the previous page.
Move to the desired page	TOC	"Table of Contents" will be displayed, enabling you to navigate to the desired page.
Exit the learning	✕	Exit the learning.

Safety precautions

When you learn based on using actual products, please carefully read the safety precautions in the corresponding manuals.

Precautions in this course

The displayed screens of the software version that you use may differ from those in this course.
This course uses the following software version:

- GX Works3 Version 1.032J

Chapter 1 | Why structured programming is necessary?

This chapter describes the necessity of structured programming for more efficient engineering.
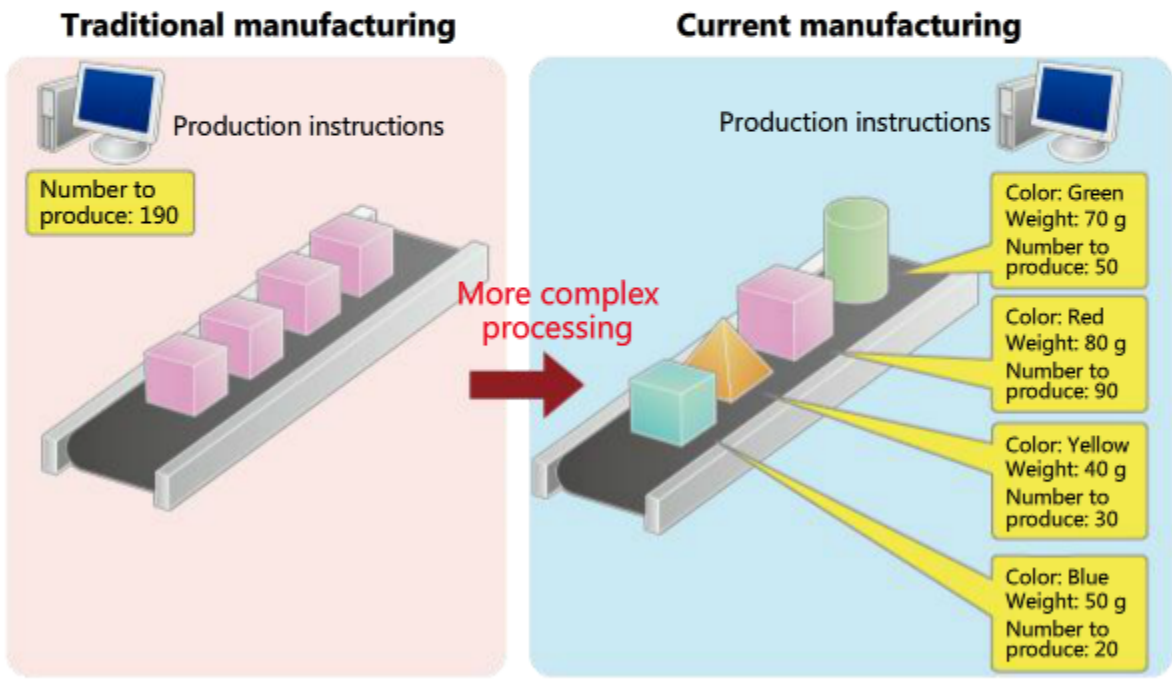
1.1 Present state of the FA industry

1.2 The necessity for programming efficiency

1.3 Efficient programming through structured programs

1.4 Structured programming with MELSOFT GX Works3

1.1 Present state of the FA industry

In the current FA industry, the data handled by manufacturing equipments is increasing with processing becoming advanced and more complex. The details are described below.
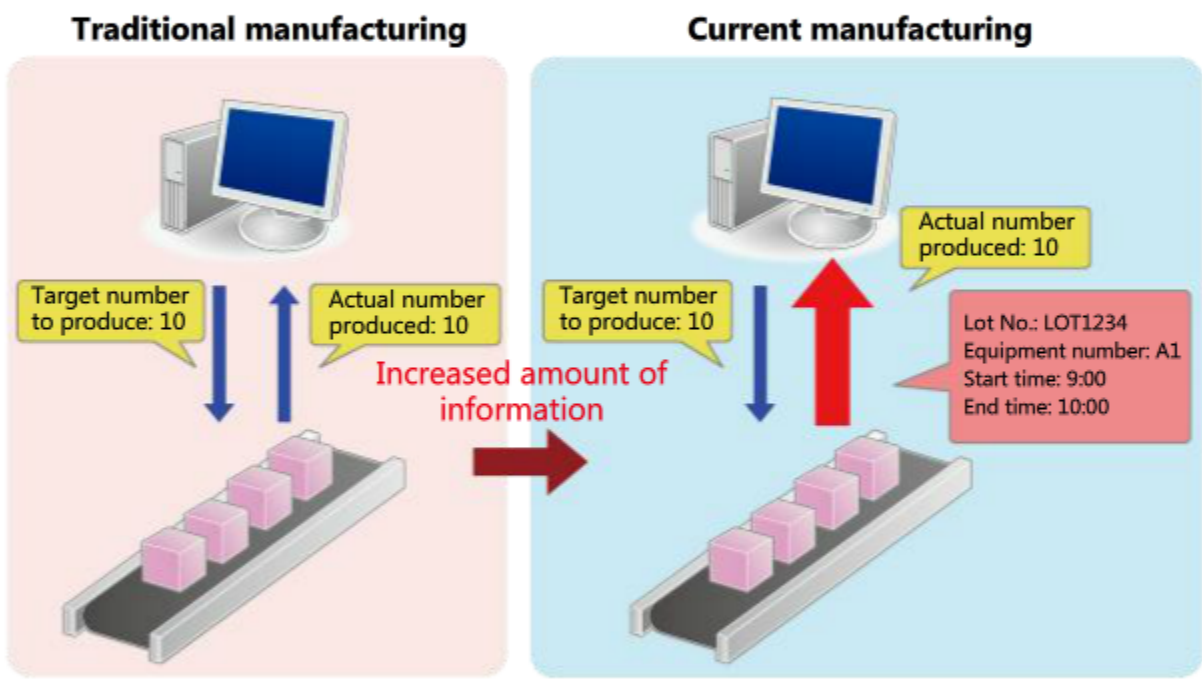
1. More complex processing

Previously, a single product was produced on one production line according to one production instruction. However, producing multiple products on one production line is becoming more main stream as a result of product diversification and streamlining of production system. For this reason, processing has become more and more complex.

Traditional manufacturing

Production instructions

Number to produce: 190

More complex processing →

Current manufacturing

Production instructions

Color: Green
Weight: 70 g
Number to produce: 50

Color: Red
Weight: 80 g
Number to produce: 90

Color: Yellow
Weight: 40 g
Number to produce: 30

Color: Blue
Weight: 50 g
Number to produce: 20

1.1 Present state of the FA industry

## 2. Increasing amounts of information

Previously, information exchanged between enterprise level computers and production floor equipment was limited to simple job instructions and production results. Nowadays, production logs are included in this production control information to ensure traceability. As such, the amount of information handled in production sites has increased.

Traditional manufacturing

Target number to produce: 10

Actual number produced: 10

Increased amount of information

Current manufacturing

Target number to produce: 10

Actual number produced: 10

Lot No.: LOT1234
Equipment number: A1
Start time: 9:00
End time: 10:00

1.1 Present state of the FA industry
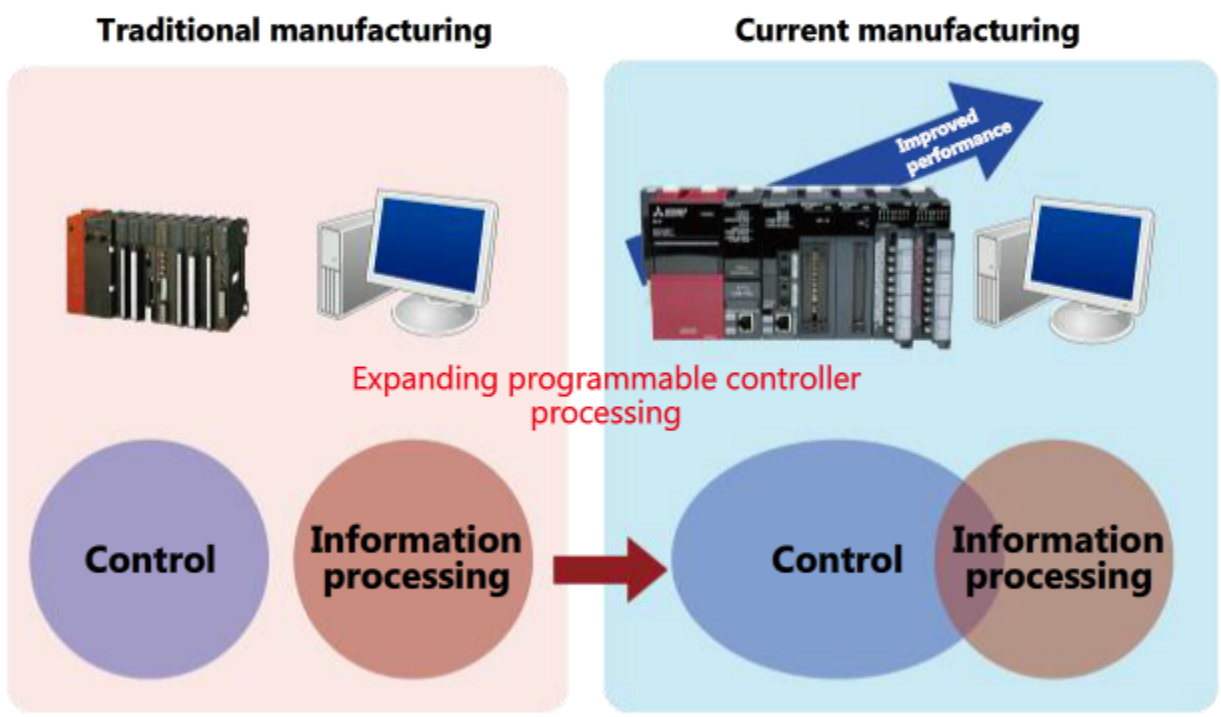
3. Large-scale system configurations

Constructing and developing large-scale factories has become increasingly common. In these large-scale factories, networks are configured for each production process and information of each process is collected by enterprise level computers. The amount of information handled in these environments is significantly greater than that in the conventional small-scale factories.
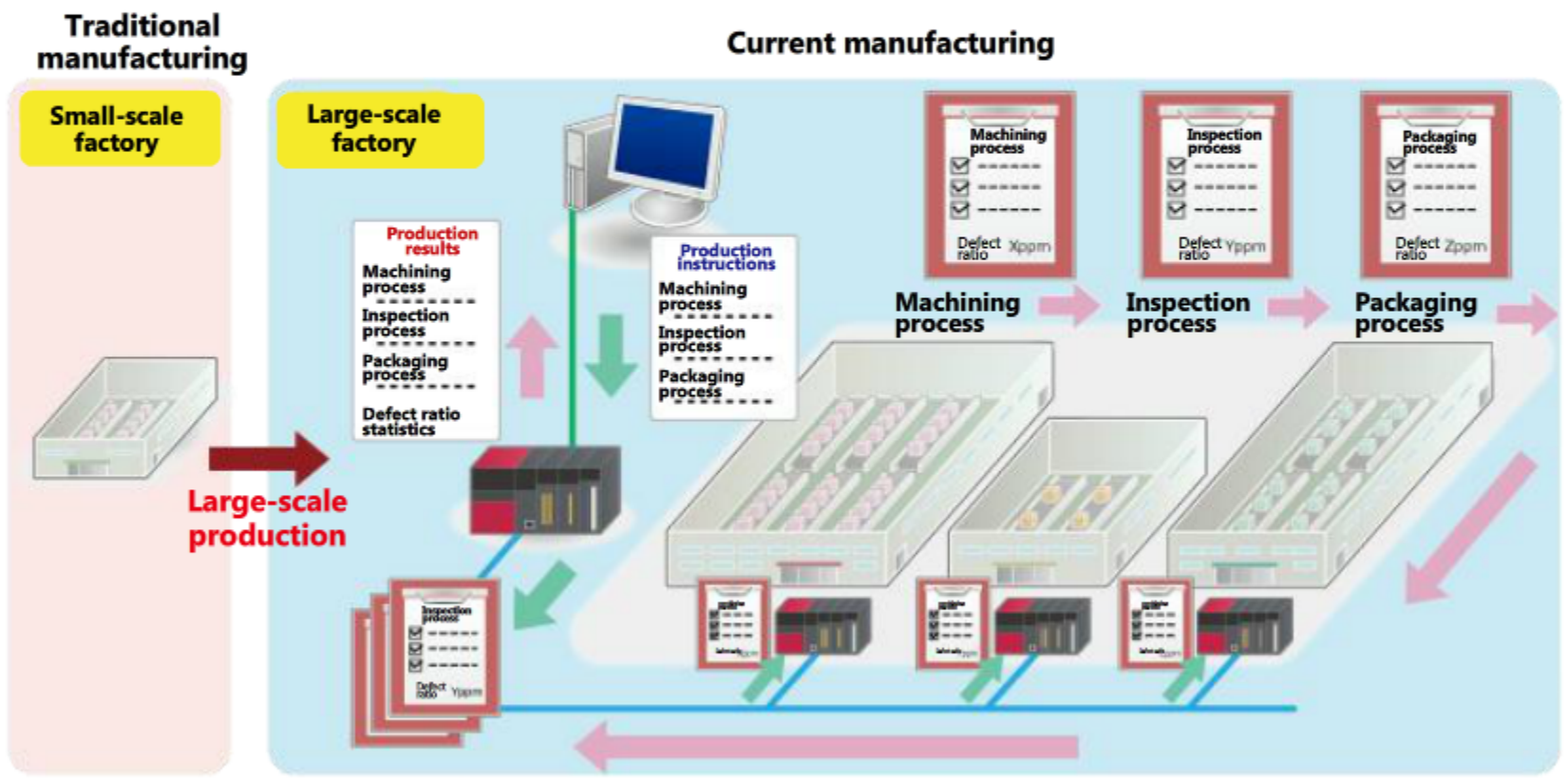
1.1 Present state of the FA industry

As information used in FA environments has increased, the types of processing performed by programmable controllers have also changed.

Previously, programmable controllers were mainly used for control while personal computers were used for information processing. Lately, both storage capacity and processing speed of programmable controllers have increased. Programmable controllers are designed to be used in FA environments and are therefore more reliable and robust than personal computers. For this reason, there has been a tendency to start relying on them for information processing.

Traditional manufacturing — **Current manufacturing**

Improved performance

Expanding programmable controller processing

Control — Information processing → Control / Information processing

1.2 The necessity for programming efficiency

As programmable controller processing has been expanding, control programs are now used for more complex processing in addition to sequential control and the number of steps has increased. Some examples of this include processing text data, performing database operations, and performing network control.

In conventional programs created using simple ladder diagrams, all processing functions are described without clear distinction. Complex programs with many steps have the following disadvantages when they are edited or new functionality is added later.

Utilizing existing programs is not easy

Determining the starting or ending of a processing is difficult.
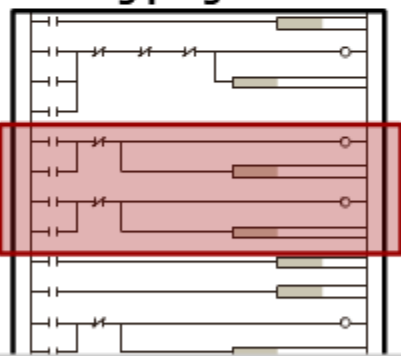Utilizable sections must be searched by referring to comments and statements.
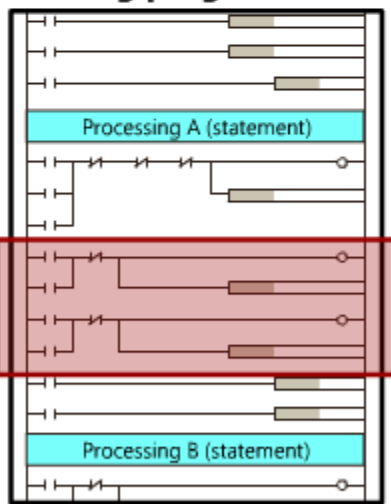
Click ◀ at the upper-right to start the animation again from the beginning.

At program development sites, efficient programming such as utilization of similar program and standardization/sharing of often used program is required to reduce engineering time for developing new programs.

1.3 Efficient programming through structured programs

Drawbacks associated with conventional, simple ladder programs are that they are difficult to understand and utilize. Structured programs can improve programming efficiency and eliminate these issues.

The following animation describes the concept of dividing programs into program parts.

Programs can be utilized by selecting individual program parts.

Existing program

New program utilizing existing programs

Processing A

Only need to select the program part
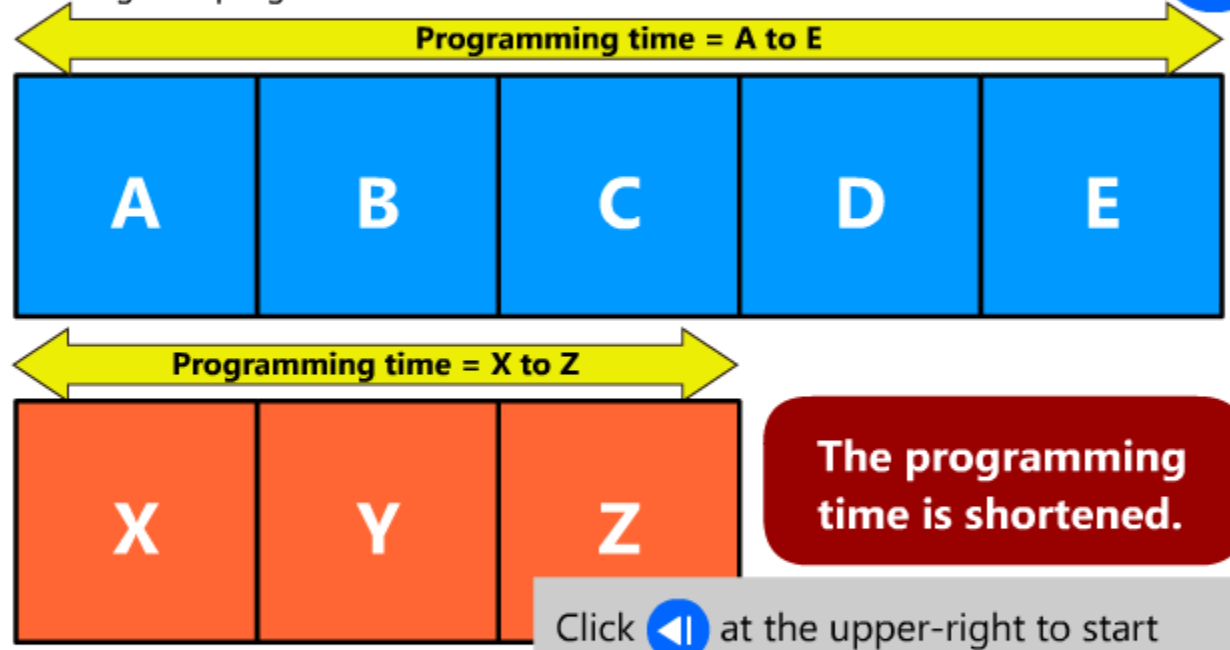
Processing B

Processing B

Processing C

Click ◀| at the upper-right to start the animation again from the beginning.

Utilization of program improves programming efficiency. The details are described in the next sections.

1.3.1 Benefits of utilizing programs

Utilization of existing programs as many as possible reduces programming time.

New program includes processing represented as B and E in the existing program. These processing can be utilized, so programming time from X through Z is only required for creating new program.

Programming time = A to E

| Existing program | A | B | C | D | E |

Programming time = X to Z

| New program | X | Y | Z |

The programming time is shortened.

Click ◀⏸ at the upper-right to start the animation again from the beginning.

In this way, utilization of existing programs will eliminate programming for new programs, reducing overall programming time. Maximizing utilization of existing programs significantly increases programming efficiency.
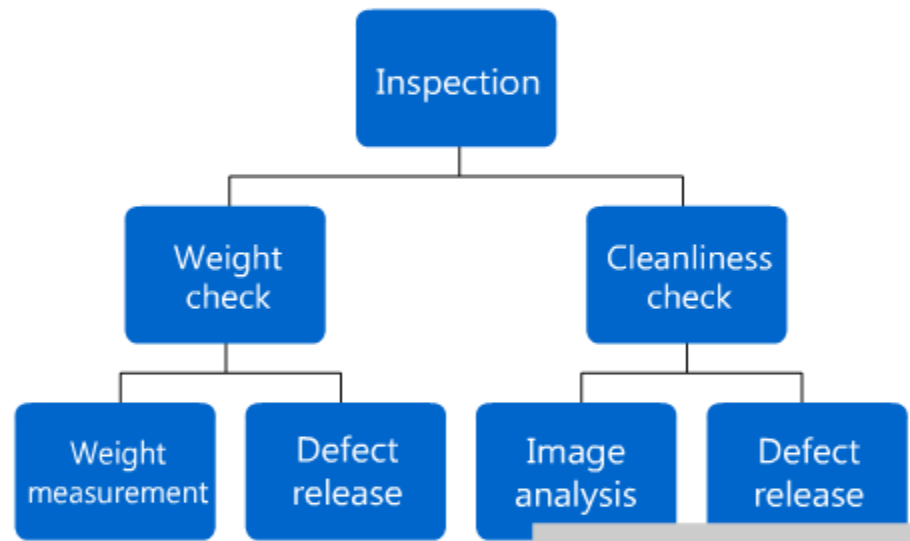
The basic concept behind creating structured programs is described in the subsequent sections.

1.3.2 Efficient selection of utilizable sections

The following animation describes the concept of hierarchical organization. Using hierarchical organization enables utilizable sections to be selected more efficiently.
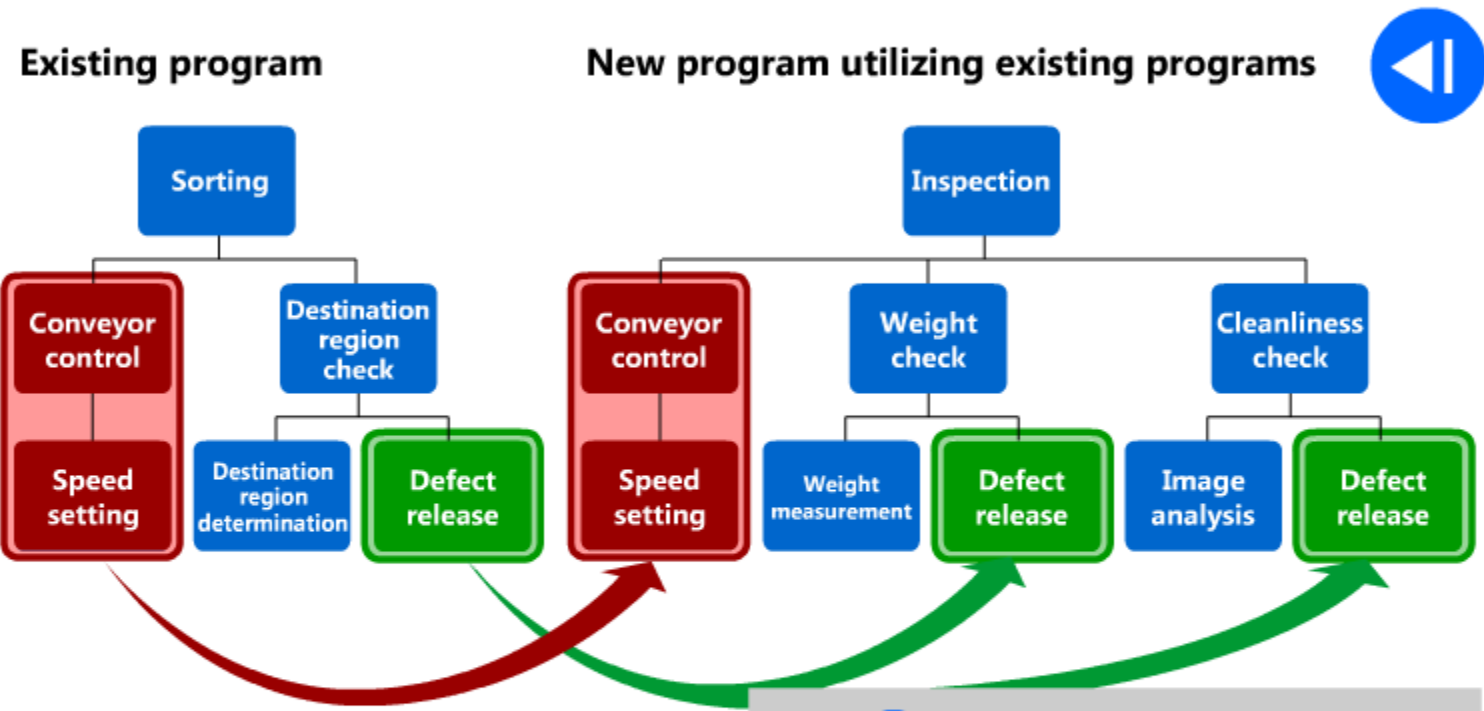
Layering POUs in this way enables programs to be viewed in smaller chunks and at a higher level at the same time, so that the utilizable sections can be selected more efficiently.

```
                    ┌─────────────┐
                    │  Inspection │
                    └─────────────┘
              ┌───────────┴───────────┐
       ┌──────────────┐        ┌──────────────┐
       │    Weight    │        │  Cleanliness │
       │    check     │        │    check     │
       └──────────────┘        └──────────────┘
        ┌──────┴──────┐         ┌──────┴──────┐
  ┌───────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐
  │  Weight   │ │  Defect   │ │  Image    │ │  Defect   │
  │measurement│ │  release  │ │ analysis  │ │  release  │
  └───────────┘ └───────────┘ └───────────┘ └───────────┘
```

Click ◀| at the upper-right to start the animation again from the beginning.

1.3.2 Efficient selection of utilizable sections

This example illustrates how efficient it can be to utilize sections from a program hierarchically organized.

Existing program

New program utilizing existing programs

Sorting

- Conveyor control
  - Speed setting
- Destination region check
  - Destination region determination
  - Defect release

Inspection

- Conveyor control
  - Speed setting
- Weight check
  - Weight measurement
  - Defect release
- Cleanliness check
  - Image analysis
  - Defect release

Utilizing multiple sections

Click ◀️ at the upper-right to start the animation again from the beginning.
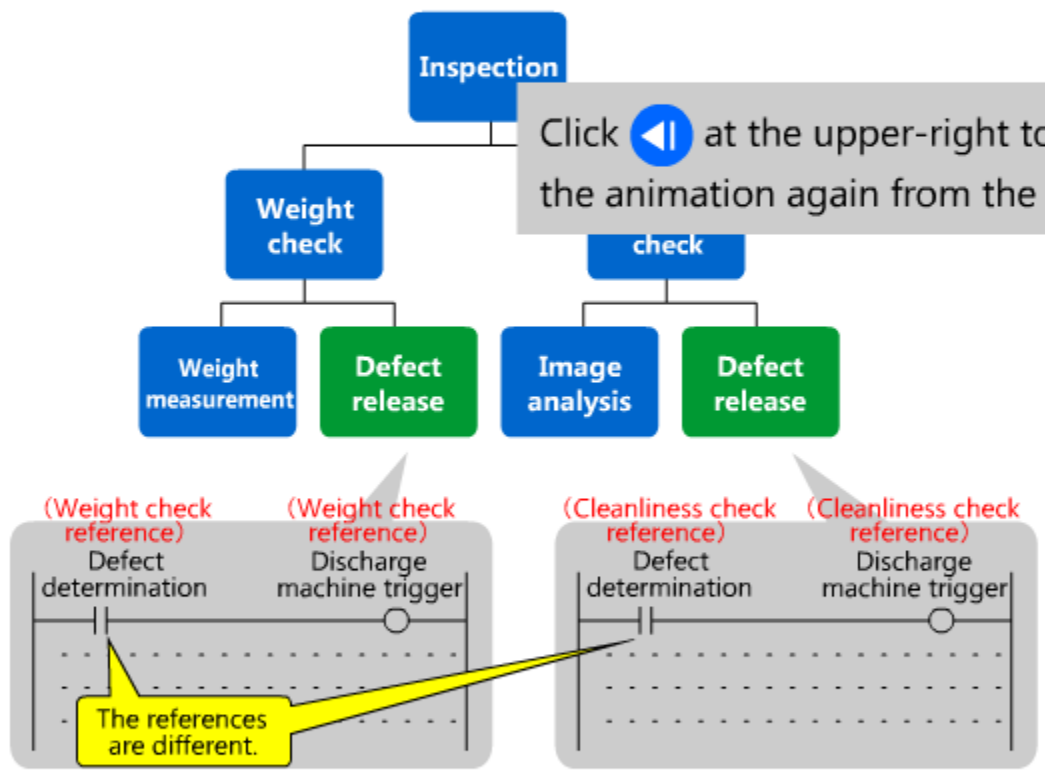
1.3.3   Utilizing programs without needing to reassign devices

You have learned how structured programs makes it easy to utilize programs. With structured programming, labels are used in place of devices. Using labels allows programmers to create programs without worrying about device conflicts with the new program.

The duplication checks for devices are not required.

Click ◀❙ at the upper-right to start the animation again from the beginning.

Inspection

Weight check

check

Weight measurement

Defect release

Image analysis

Defect release

(Weight check reference)
Defect determination

(Weight check reference)
Discharge machine trigger

(Cleanliness check reference)
Defect determination

(Cleanliness check reference)
Discharge machine trigger
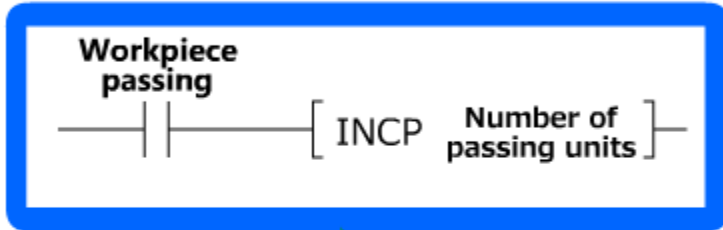
The references are different.

1.3.4 Programming using labels for specific processing

Using labels enables programs to be created without having to deal with comments and devices.

Using labels enables programs to be created with the names of processing without having to worry about devices.

Using labels

Workpiece passing

—| |——[INCP   **Number of passing units**]—

Intuitive and easy to understand

Click ◄‖ at the upper-right to start the animation again from the beginning.

1.4 Structured programming with MELSOFT GX Works3

The engineering software MELSOFT GX Works3 supports creation of structured programs.

MELSOFT GX Works3 makes it easy to utilize programs by displaying the available program organization units (POUs) for easy selection.
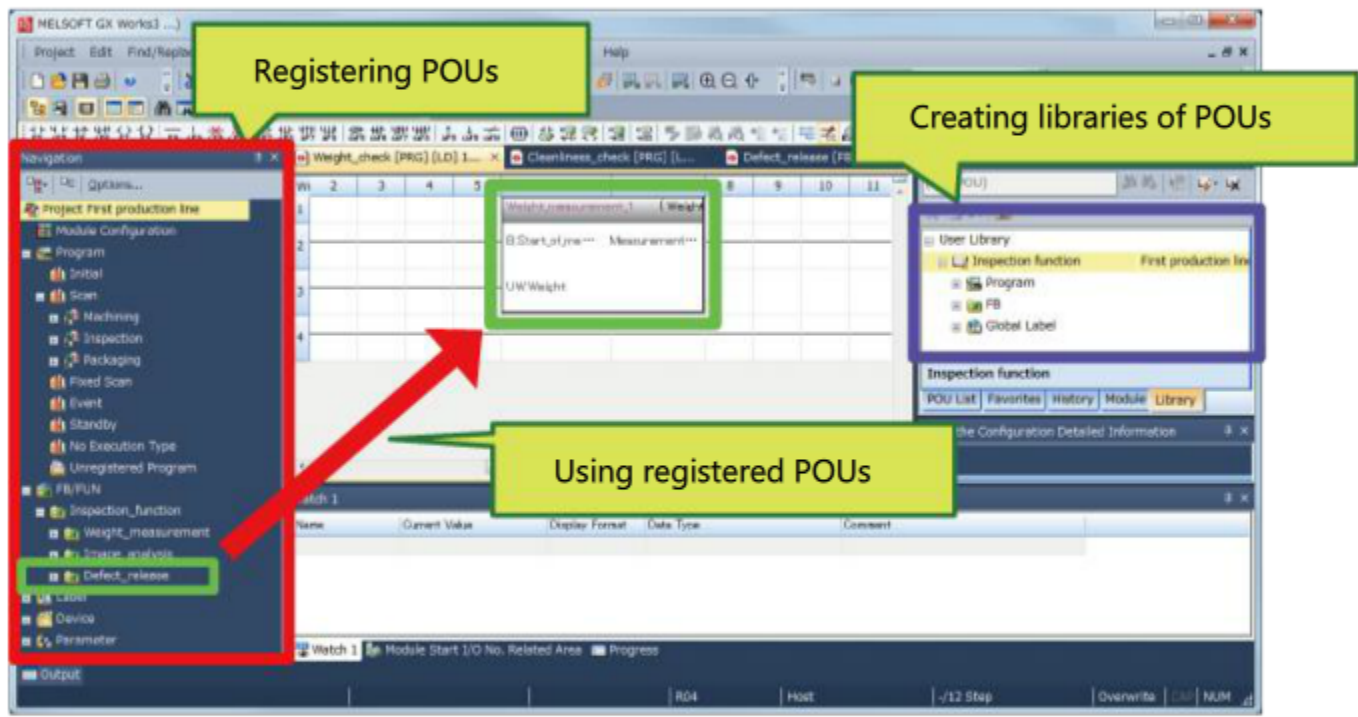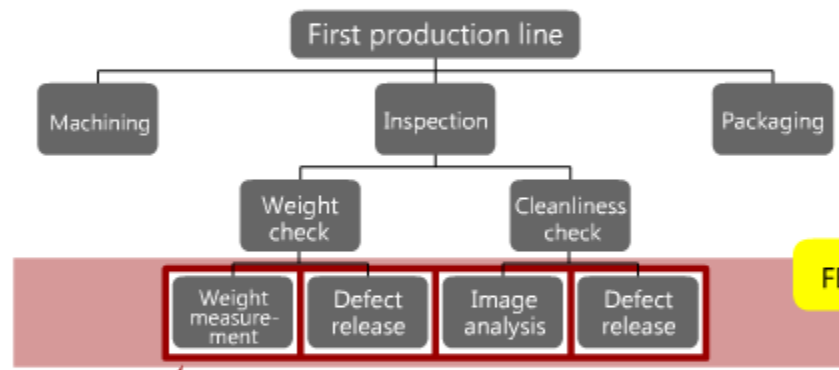

MELSOFT GX Works3

1.4.1 How hierarchical organization of programs are represented in GX Works3

The following animation describes how GX Works3 represents the hierarchical organization of programs, using an example of production line.

Weight check processing includes the weight measurement and defect release functions.
The weight measurement, image analysis, and defect release functions included in the processing are represented as lower levels under the FB/FUN level.

First production line
- Machining
- Inspection
  - Weight check
    - Weight measure-ment
    - Defect release
  - Cleanliness check
    - Image analysis
    - Defect release
- Packaging

Function

Names of FB and FUN can be assigned by users.

FB/FUN

Project First production line
- Module Configuration
- Program
- FB/FUN
  - Inspection_function
    - Weight_measureme
      - Local Label
      - ProgramBody
    - Image_analysis
      - Local Label
      - ProgramBody
    - Defect_release
      - Local Label
      - ProgramBody

Click ◀| at the upper-right to start the animation again from the beginning.

1.4.2 Using program organization units (POUs)

This section explains the procedure to use POUs in GX Works3. FB/FUN are selected from the project window and arranged in the program.

Project First production line
Module Configuration
Program
  Initial
  Scan
    Machining
    Inspection
      Weight_check
        Local Label
        ProgramBody
      Cleanliness_check
    Packaging
  Fixed Scan
  Event
  Standby
  No Execution Type
  Unregistered Program
FB/FUN
  Inspection_function
    Weight_measurement
      Local Label
      ProgramBody
    Image_analysis
    Defect_release

Inspection

Weight check

Cleanliness check

Weight measurement

Defect release

Image analysis

Defect release

1	2	3	4	5	6	7	8	9	10	11	12

Weight_measurement_1   (Weight_measurement)

B:Start_of_measurement   Measurement_result:UW

UW:Weight

END

1	2	3	4	5	6	7	8	9	10	11	12
<	Weight	K100							MOV	K1	Measurement_result
=	Weight	K100									
>	Weight	K100									

Click ◀‖ at the upper-right to start the animation again from the beginning.

POUs can be shared among projects using the library function. Creating libraries of proven POUs helps ensure consistent quality of programs.
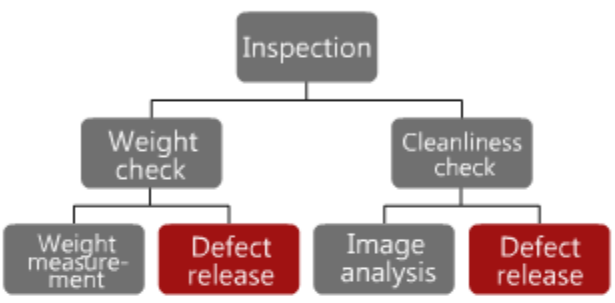
POUs can be used by simply dragging and dropping them onto the program.
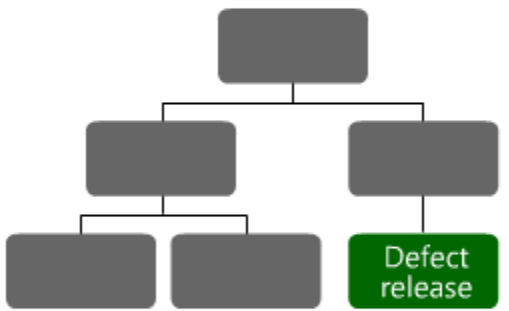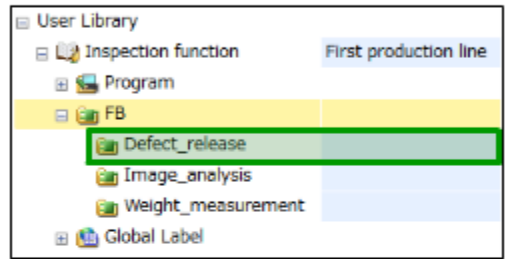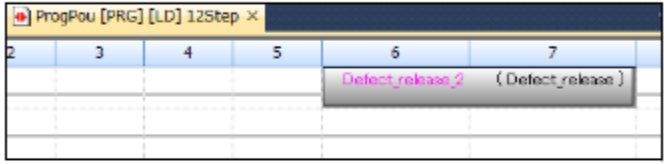
⊕ ProgPou [PRG] [LD] 12Step ✕						
2	3	4	5	6	7	
				Defect_release_2	(Defect_release)	

User Library	First production line
⊟ 📖 Inspection function	
⊟ 🖥 Program	
⊟ 📦 FB	
📦 Defect_release	
📦 Image_analysis	
📦 Weight_measurement	
⊕ 🏷 Global Label	

Defect release
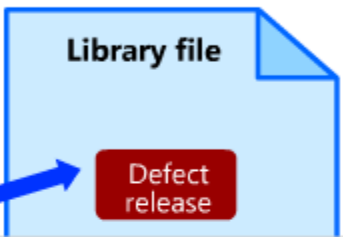
Click ◀| at the upper-right to start the animation again from the beginning.

It is particularly effective to create libraries of POUs often utilized. In the following inspection programs, the "defect release" FB/FUN is the most appropriate for registering to the library.

Library file

Often-used POUs are registered.

Defect release

Inspection
- Weight check
  - Weight measurement
  - Defect release
- Cleanliness check
  - Image analysis
  - Defect release

Click ◀ at the upper-right to start the animation again from the beginning.

Equipment A

Sorting
- Conveyor Control
  - Speed setting
- Defect check
  - Error detection
  - Defect release
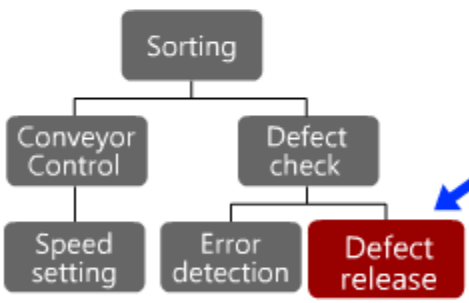
Equipment B

Utilize   Utilize   Utilize   Utilize
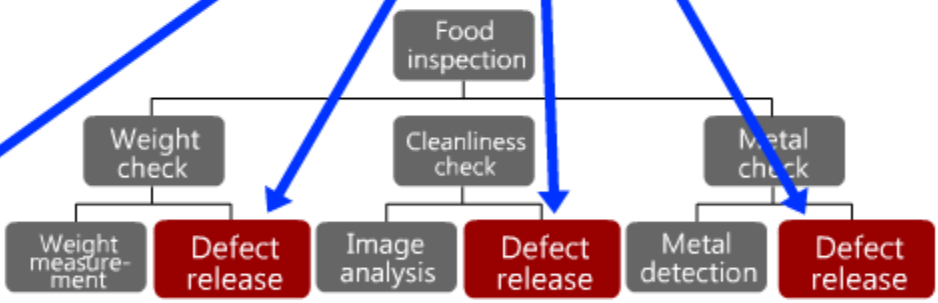
Food inspection
- Weight check
  - Weight measurement
  - Defect release
- Cleanliness check
  - Image analysis
  - Defect release
- Metal check
  - Metal detection
  - Defect release

1.4.4 Using pre-made POUs

GX Works3 is pre-configured with POUs for many typical programs. These pre-made POUs include module FBs that are automatically registered in accordance with the module configuration and the MELSOFT Library.

The MELSOFT Library provided by Mitsubishi Electric has many useful FBs available for users. The FBs help you use the modules manufactured by Mitsubishi Electric or products of other manufactures.
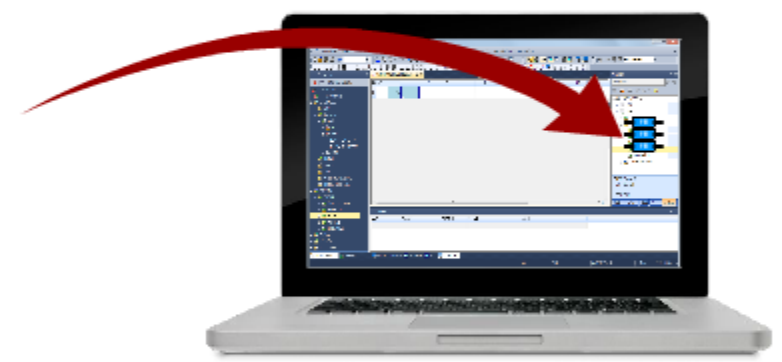
Click ◀❚ at the upper-right to start the animation again from the beginning.

For the MELSOFT Library, please contact your local Mitsubishi Electric representative.

1.4.5 Efficient entry of labels

The predictive text feature helps you enter labels more efficiently.
When you begin entering a label, a list of registered labels that match the first few characters entered appears automatically.
Then the desired label is entered just by selecting. This enables you to place the same label at multiple locations without any entry mistakes.

Write		1	2	3	4	5	6	7	8	9
1		Start_conve···								
2	(10)									

Click ◀❙ at the upper-right to start the animation again from the beginning.

Programming using registered labels

1.5   Summary

The contents of this chapter are:

- Present state of the FA industry

- Drawbacks of programs using simple Ladder

- Concepts and advantages of structured programs

- Structured programming with MELSOFT GX Works3

Important points to consider:

Benefits of utilizing programs	Utilizing programs reduces the amount of new development required. Maximizing utilization of existing programs significantly increases programming efficiency.
Dividing programs into program parts	Utilizable sections can be clarified by dividing programs into program parts and assigning meaningful names to these parts.
Hierarchical organization	In addition to dividing programs into program parts, using a hierarchical organization for programs makes it easier and more efficient to select utilizable sections of programs.
Advantages of using labels	• Using labels eliminates the need for device conflict checks and device reassignment, which makes utilizing programs more efficient • Programming is more intuitive when meaningful names are used to represent specific processing
Structured programs in MELSOFT	• MELSOFT enables the creation of structured programs in an easily understandable visual editor • MELSOFT helps improve programming efficiency

Chapter 2   Why structured programs are so efficient

This chapter describes the benefits of structured programming.

The programming process will be described as divided into the following phases.

- Program editing: Creating and editing the program

- Evaluation testing: Confirming that the program functions correctly

- Search and fix: Identifying and fixing bugs

2.1 Shortening programming time by utilizing programs

2.2 Removing the device reassignment process
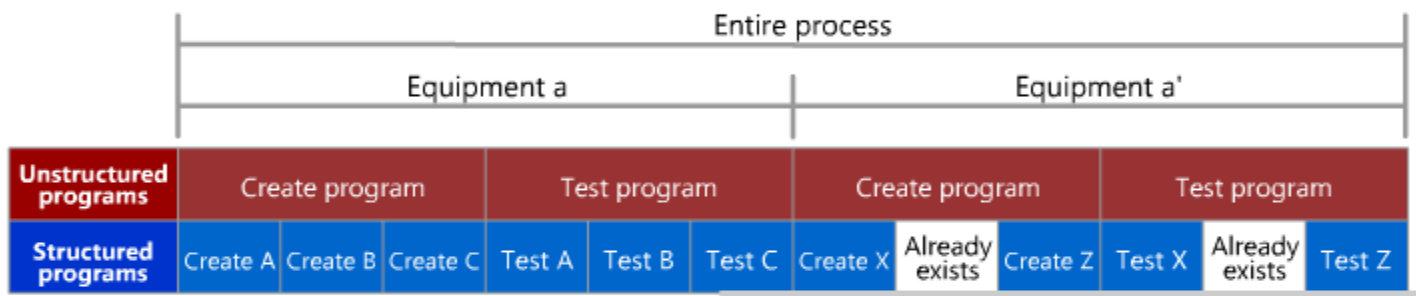
2.3 Preventing accidental changes

2.1 Shortening programming time by utilizing programs

Utilizing existing programs shortens the programming time and improves programming efficiency.

Efficient programming using structured programs

Programming process

	Entire process								
	Equipment a						Equipment a'		
Unstructured programs	Create program			Test program			Create program		Test program
Structured programs	Create A	Create B	Create C	Test A	Test B	Test C	Create X	Already exists	Create Z	Test X	Already exists	Test Z

Click ◀ at the upper-right to start the animation again from the beginning.

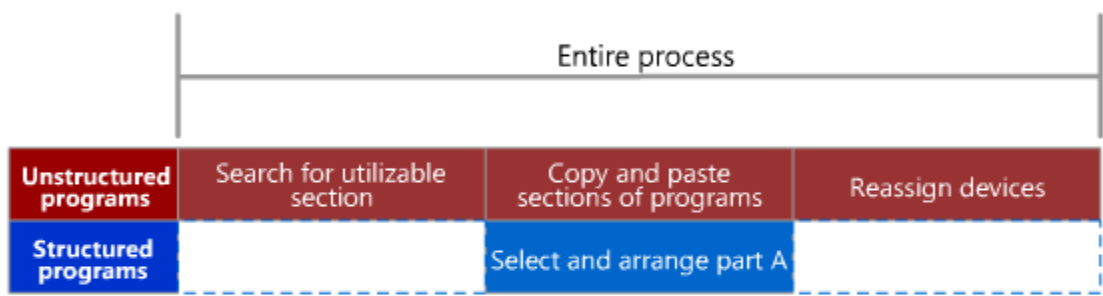| Create A	Create B	Create C	Test A	Test B	Test C	Create X	Create Z	Test X	Test Z	Removed

2.2 Removing the device reassignment process

Utilizing structured programs removes the need to reassign devices in the new program.

Efficient programming using structured programs

Programming process

Entire process			
Unstructured programs	Search for utilizable section	Copy and paste sections of programs	Reassign devices
Structured programs		Select and arrange part A	

Select and arrange part A	Removed

Click ◀ at the upper-right to start the animation again from the beginning.

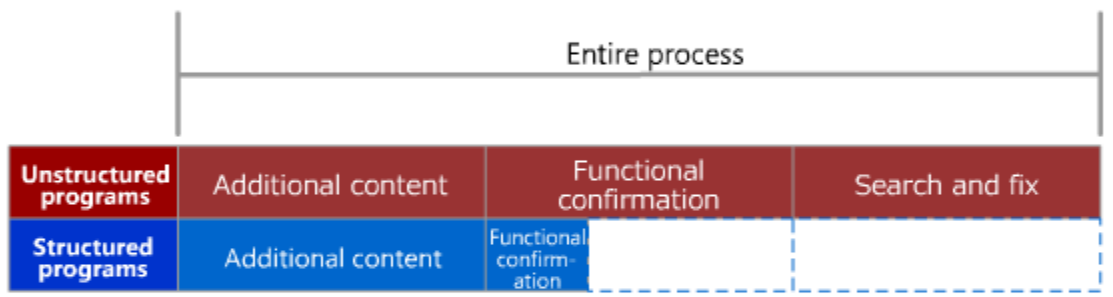## 2.3 Preventing accidental changes

Accidental changes that could result from adding functionality to programs can be prevented by using structured programs.

Efficient programming using structured programs

Programming process

	Entire process		
Unstructured programs	Additional content	Functional confirmation	Search and fix
Structured programs	Additional content	Functional confirm-ation	

Additional content	Functional confirm-ation	Unnecessary

Click ◀ at the upper-right to start the animation again from the beginning.

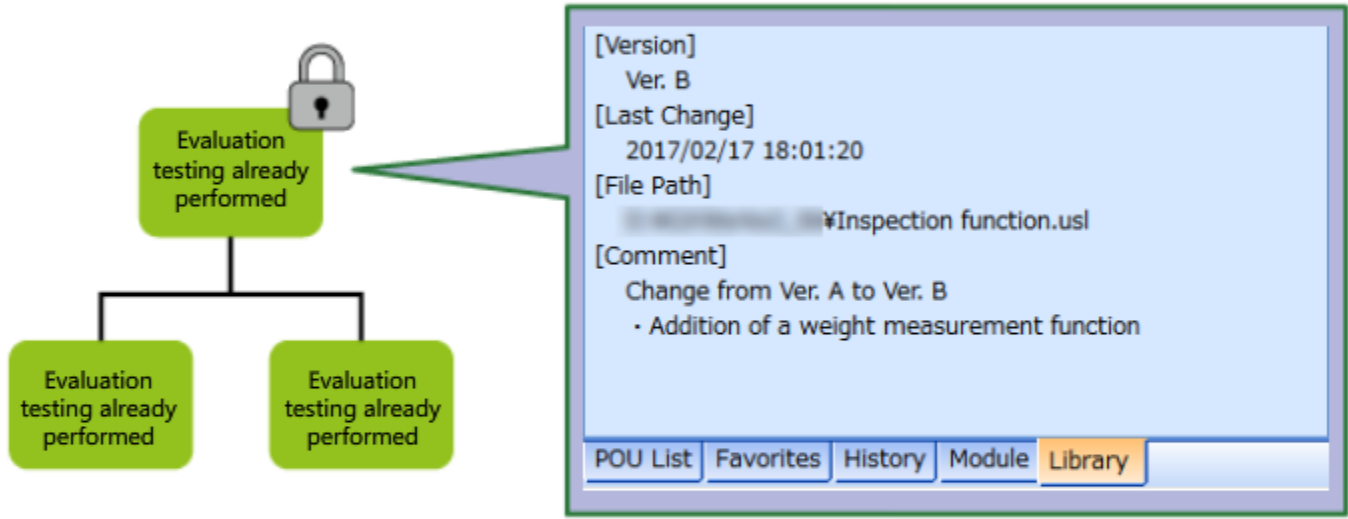**Secure management of POUs**

POUs can be managed securely using MELSOFT GX Works3.

Accidental changes can be completely eliminated by locking individual POU. Each POU can have its own information including the version, the date of update, and comments. Change history can be logged in comments.

Evaluation testing already performed

Evaluation testing already performed

Evaluation testing already performed

[Version]
  Ver. B
[Last Change]
  2017/02/17 18:01:20
[File Path]
  ¥Inspection function.usl
[Comment]
  Change from Ver. A to Ver. B
   ・Addition of a weight measurement function

POU List | Favorites | History | Module | Library

2.4 Summary

The contents of this chapter are:

- Benefits of utilizing POUs

- Benefits of using labels

- Benefits of using structured programs

Important points to consider:

Reduced evaluation testing	Using structured programs reduces not only the time required to create programs, but also the time required for evaluation testing.
Preventing accidental changes	With structured programs, changes are made to individual POU, which prevents accidental changes made to other program parts.
Locking POUs	Locking a POU once evaluation testing is complete ensures secure management of POUs.

2.5 Course summary

This course described the use of structured programming to make programming more efficient. You have now completed this course.

Here is a summary of what was covered in this course.

- Structured programming is a method of efficiently creating programs that have become increasingly complex along with manufacturing trends of large-scale systems and increased data amount

- Structured programs are easier to utilize which reduces the amount of new development and improves programming efficiency

- Dividing programs into program parts enables programs to be divided by processing and function. Using a hierarchical organization and meaningful names for program parts clarify the sections of programs that can be utilized.

- Using hierarchical organization enables utilizable sections to be selected more efficiently

- Using labels allows POUs to be utilizable without having to worry about device conflicts with the new program

- MELSOFT GX Works3 supports structured programming and improves programming efficiency

- Creating libraries enables sharing/standardization of often-used program parts, realizing consistent quality of programs

- In addition to utilizing programs, existing libraries can be utilized to reduce the amount of new development

When actually implementing structured programming, refer to the following.

Programming with MELSOFT GX Works3

- e-Learning course "GX Works3 (Ladder)"
- Operations manuals for MELSOFT GX Works3

Program structure and syntax

- Programming manuals for MELSEC iQ-R Series

Test   Final Test

Now that you have completed all of the lessons of the PLC Efficient Programming course, you are ready to take the final test. If you are unclear on any of the topics covered, please take this opportunity to review those topics.
There are a total of 9 questions (10 items) in this Final Test.
You can take the final test as many times as you like.

How to score the test
After selecting the answer, make sure to click the **Answer** button. Your answer will be lost if you proceed without clicking the Answer button. (Regarded as unanswered question.)

Score results
The number of correct answers, the number of questions, the percentage of correct answers, and the pass/fail result will appear on the score page.

Correct answers :   **9**

Total questions :   **9**

To pass the test, you have to answer **60%** of the questions correct.

Percentage :   **100%**
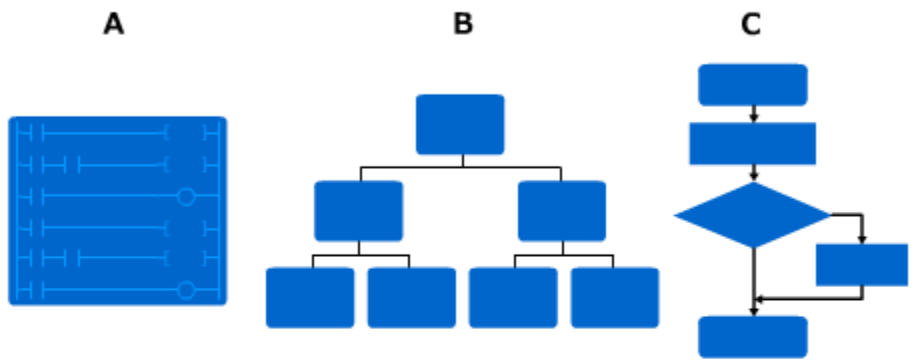
[Continue]   [Review]

- Click the **Proceed** button to exit the test.
- Click the **Review** button to review the test. (Correct answer check)
- Click the **Retry** button to retake the test again.

Test   Final Test 1

✅ Structured programs Select the correct illustration of structured programs.

- ⭘ A
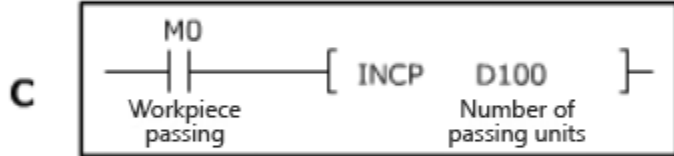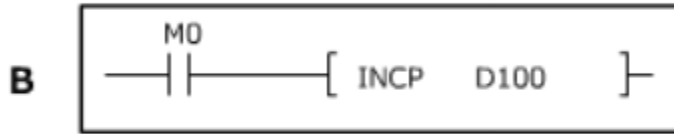- ☑ ⦿ B
- ⭘ C

A

B

C

Back

Test   Final Test 2

✅ Using labels Select the program created using labels.

- ☑ ⦿ A
- ○ B
- ○ C

A


```
Workpiece
passing                     Number of
  ─┤├─          [ INCP    passing units ]─
```

B


```
  M0
  ─┤├─          [ INCP    D100 ]─
```

C


```
  M0
  ─┤├─          [ INCP    D100 ]─
Workpiece                   Number of
 passing                   passing units
```
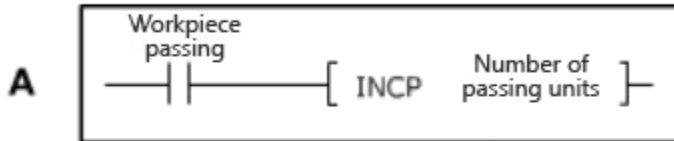
Back

✅ Process of utilizing programs Select the flowchart that correctly illustrates the process of utilizing structured programs created using labels.
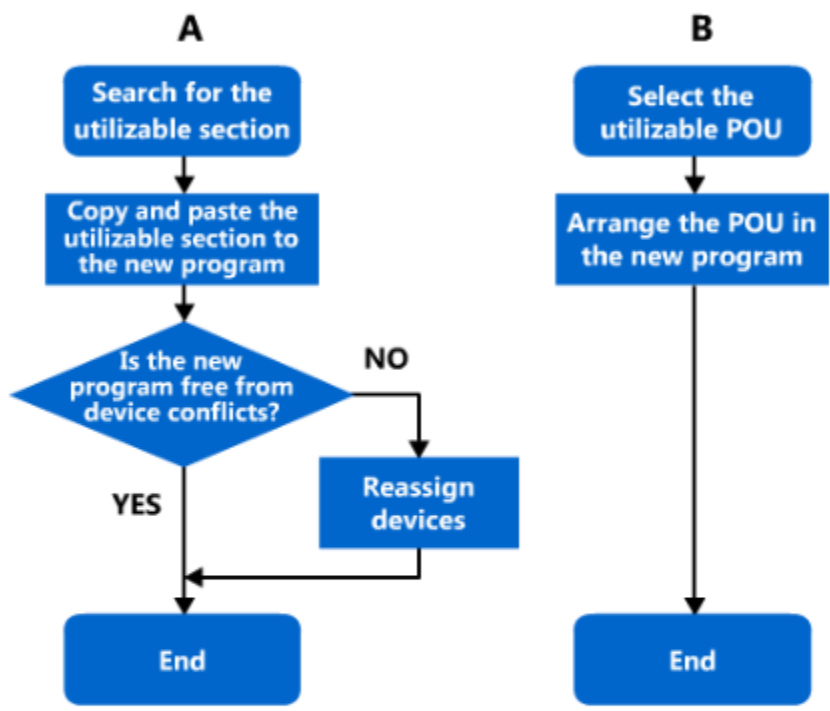
○ A

✅ ◉ B

A

Search for the utilizable section
↓
Copy and paste the utilizable section to the new program
↓
Is the new program free from device conflicts? —NO→ Reassign devices
YES ↓ ←
End

B

Select the utilizable POU
↓
Arrange the POU in the new program
↓
End

Back

✅ Purpose of structured programs Select the correct description about the purpose of structured programs. (Multiple answers)

- ☐ Improve program processing speed
- ✅ ☑ Improve programming efficiency
- ☐ Reduce program compiling time
- ✅ ☑ Improve visualization of programs

Back

✅ Structured programming in GX Works3 Select the correct description about creating structured programs in GX Works3.

☑ 🔘 Structured programs can be created immediately after simply installing the software.

⚪ Creating structured programs in GX Works3 requires the purchase of a plug-in.

Back

✅ Dividing programs into program parts Select all statements that describe the dividing of programs into program parts. (Multiple answers)

- ☐ Divide programs in a predetermined, equal number of steps
- ☑ ☑ Divide programs into individual processing and functions
- ☑ ☑ Assign meaningful names to processes
- ☐ Use pointers to jump to different sections
- ☑ ☑ Program organization units (POUs) refer to the result of dividing programs into program parts

Back

Test | Final Test 7

✅ Advantages of libraries Select the correct description about the advantages of registering programs into libraries. (Multiple answers)

- ☑ ✔ Registering often-used POUs enables efficient utilization of programs
- ☐ Prevents the utilization of POUs
- ☑ ✔ POUs can be shared between many people
- ☑ ✔ Registering and reusing standardized programs ensures consistent program quality

Back

✅ Utilizing libraries Module FBs and MELSOFT Library Select the correct description about the utilization of libraries such as module function blocks and the MELSOFT Library.

☑ 🔘 Internal operation does not need to be verified when pre-made libraries are used

⚪ Module FBs must be created in accordance with the module model

Back

✅ Pre-configured Libraries Select the option that represents the library provided by Mitsubishi Electric.

Q1 Module FBs
Q2 MELSOFT Library

Q1 Libraries automatically prepared in accordance with module model configurations ▼
    ✅

Q2 Libraries created from programs for connecting vision sensors and laser displacement sensors ▼
    ✅

Back

Test | Test Score

You have completed the Final Test. You results area as follows.
To end the Final Test, proceed to the next page.

Correct answers : **9**

Total questions : **9**

Percentage : **100%**

[Continue]   [Review]

Congratulations. You passed the test.

You have completed the PLC Efficient Programming course.

Thank you for taking this course.

We hope you enjoyed the lessons and the information you acquired in
this course will be useful in the future.

You can review the course as many times as you want.

Review   Close