Introduction to Flowcharting

A Supplement to

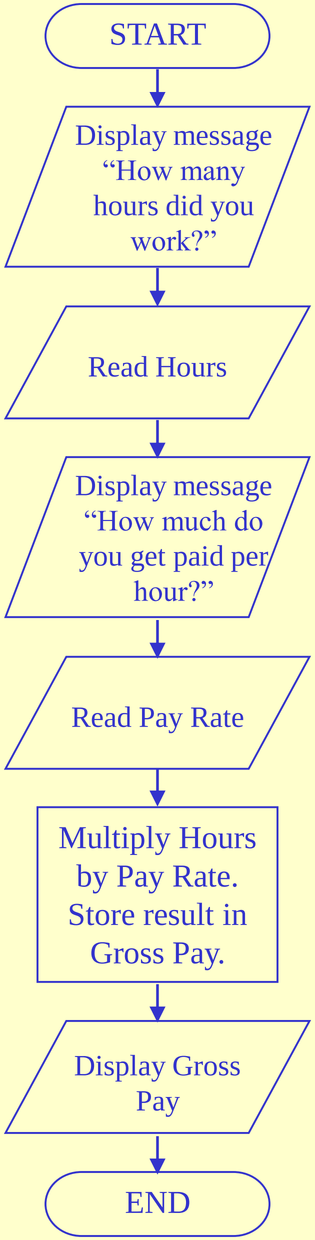
Starting Out with C++, 4th Edition

by Tony Gaddis

Published by Addison-Wesley

What is a Flowchart?

- A flowchart is a diagram that depicts the “flow” of a program.
- The figure shown here is a flowchart for the pay-calculating program in Chapter 1.



START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

Read Pay Rate

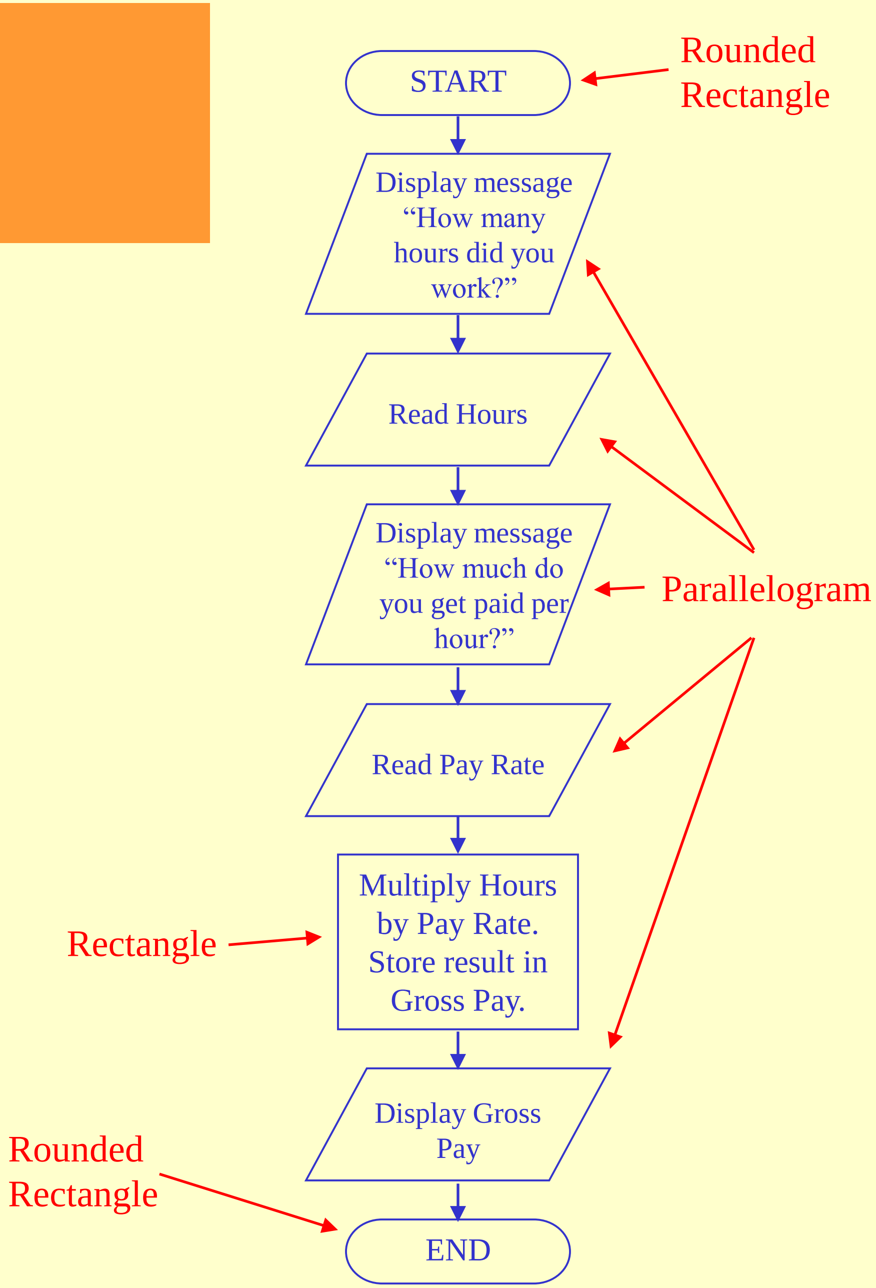
Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

END

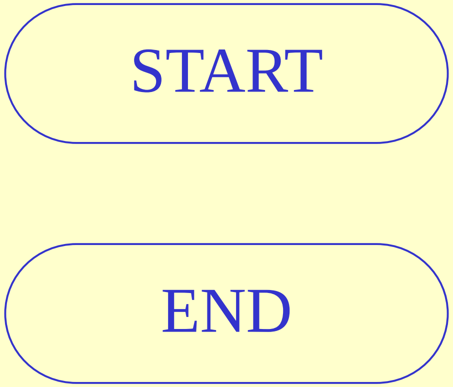
Basic Flowchart Symbols

- Notice there are three types of symbols in this flowchart:
  - rounded rectangles
  - parallelograms
  - a rectangle
- Each symbol represents a different type of operation.


Rounded Rectangle
START
Display message "How many hours did you work?"
Read Hours
Display message "How much do you get paid per hour?"
Parallelogram
Read Pay Rate
Rectangle
Multiply Hours by Pay Rate. Store result in Gross Pay.
Display Gross Pay
Rounded Rectangle
END

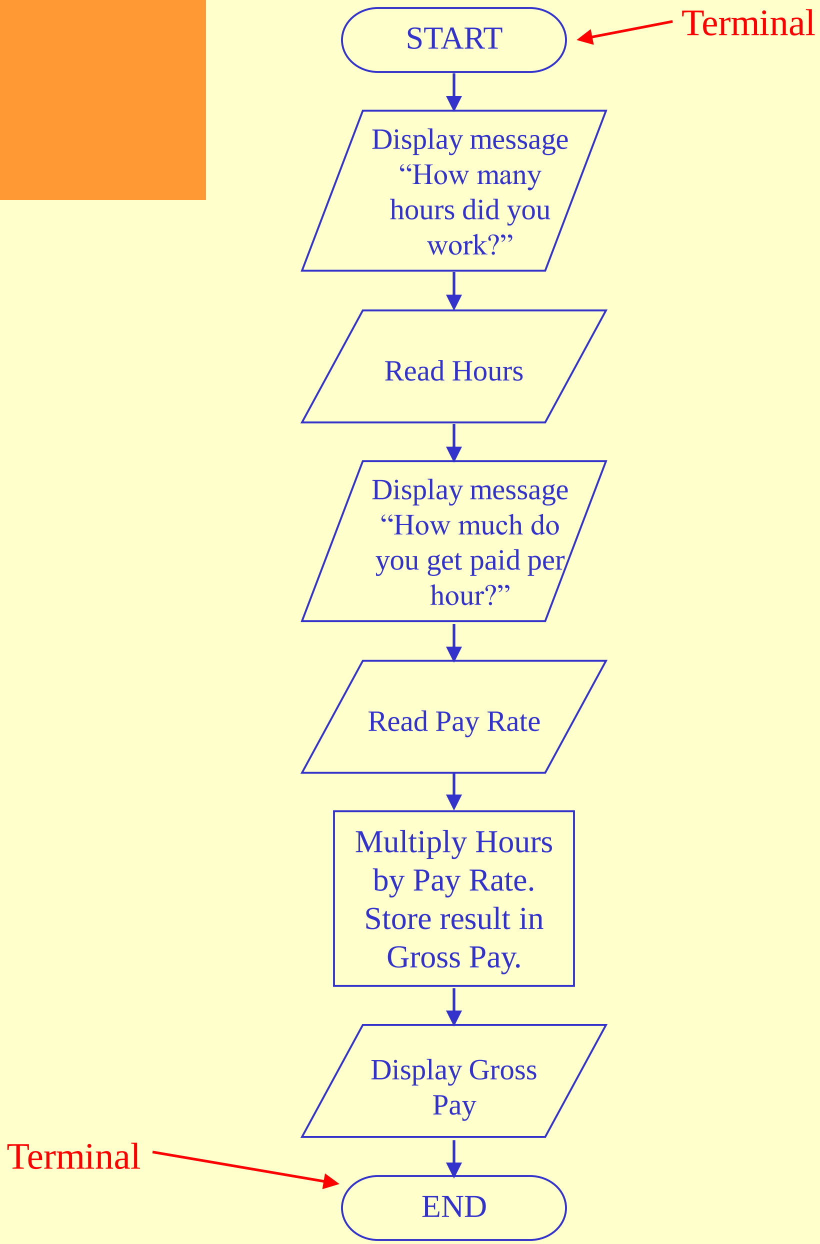
Basic Flowchart Symbols

- Terminals
  - represented by rounded rectangles
  - indicate a starting or ending point



START

END



Terminal

START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

Read Pay Rate

Multiply Hours by Pay Rate. Store result in Gross Pay.

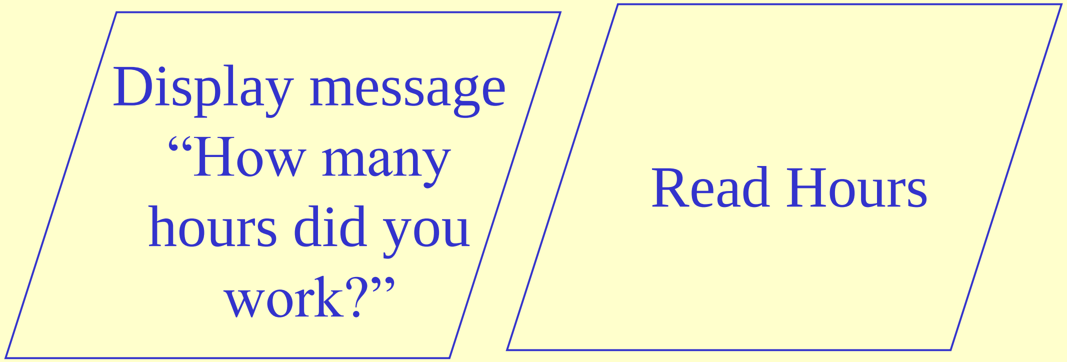
Display Gross Pay

Terminal

END

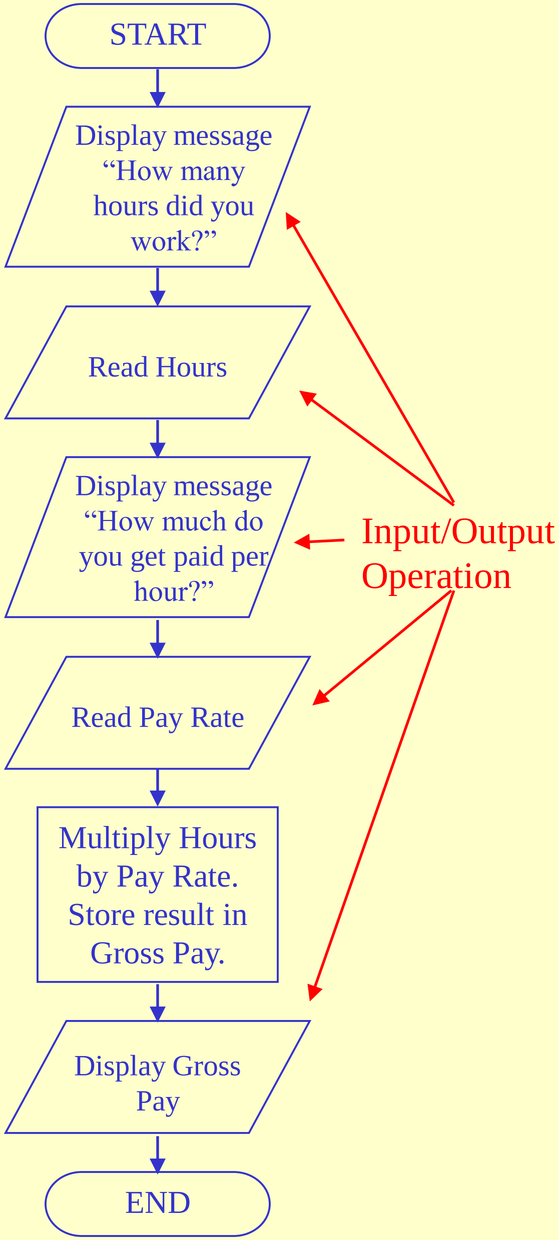
Basic Flowchart Symbols

- Input/Output Operations
  - represented by parallelograms
  - indicate an input or output operation



Display message “How many hours did you work?”

Read Hours



START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

Read Pay Rate

Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

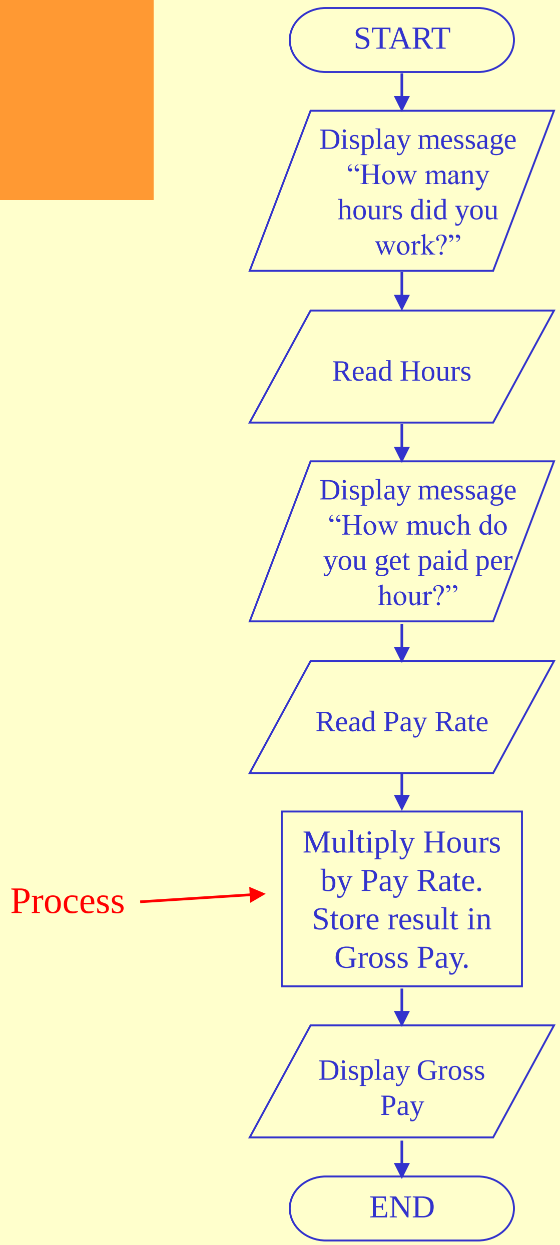
END

Input/Output Operation

Basic Flowchart Symbols

- Processes
  - represented by rectangles
  - indicates a process such as a mathematical computation or variable assignment

Multiply Hours by Pay Rate. Store result in Gross Pay.



Process

START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

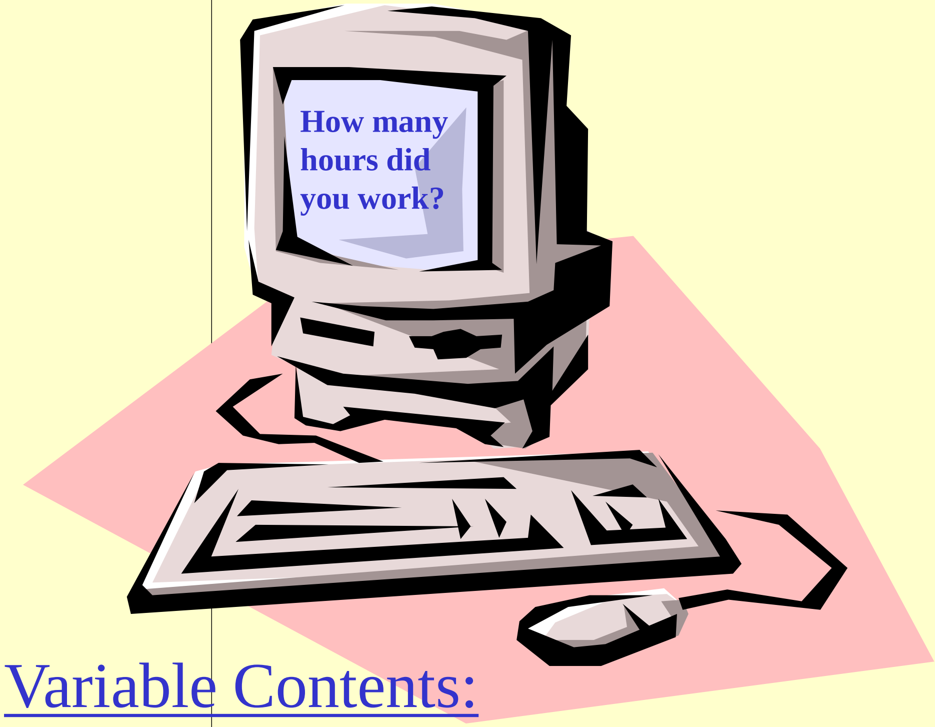
Read Pay Rate

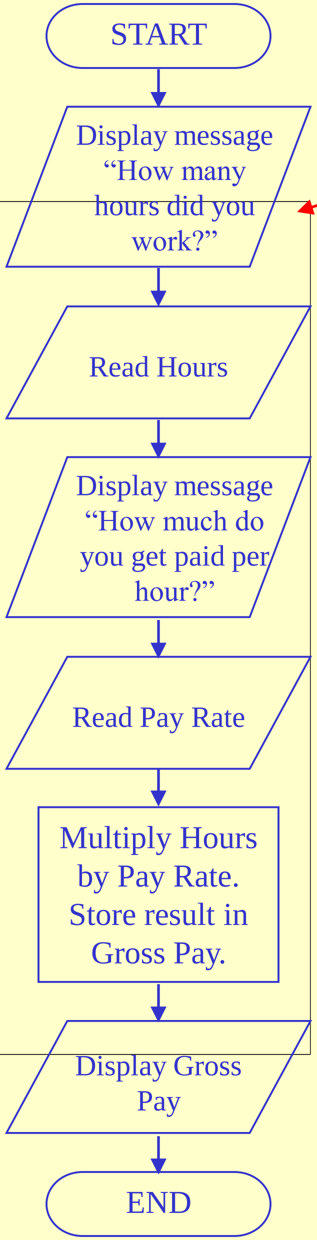
Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

END

Stepping Through the Flowchart




START

Display message "How many hours did you work?"

Read Hours

Display message "How much do you get paid per hour?"

Read Pay Rate

Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

END

Output Operation

Variable Contents:

Hours: ?

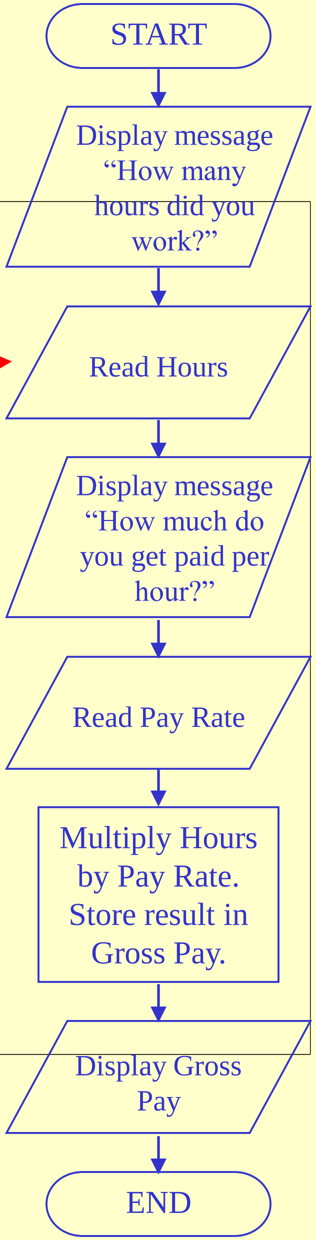
Pay Rate: ?

Gross Pay: ?

Stepping Through the Flowchart

Input Operation

(User types 40)



START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

Read Pay Rate

Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

END

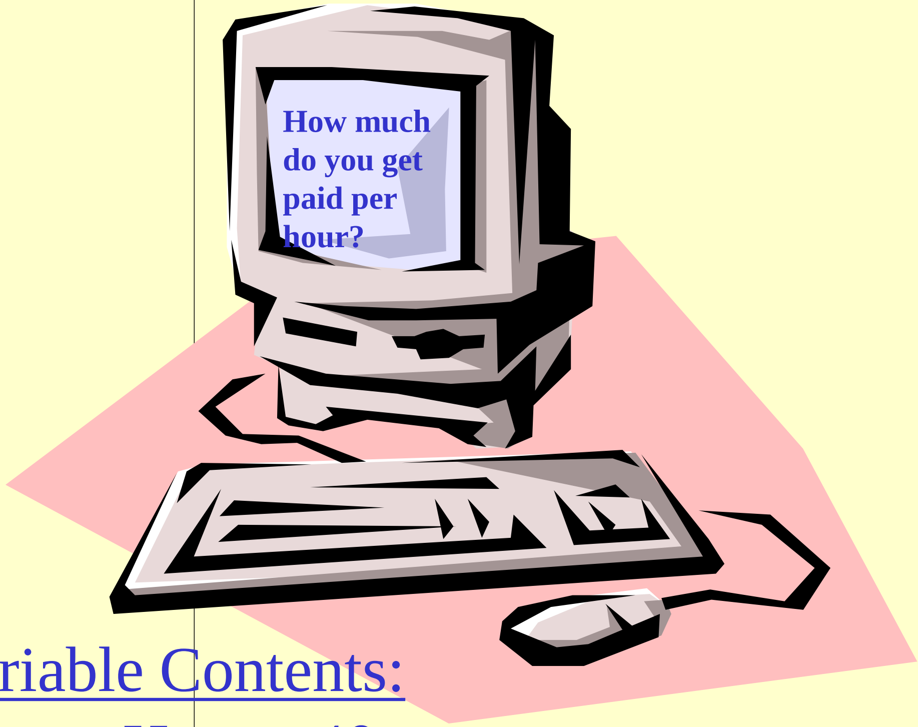
Variable Contents:

Hours: 40

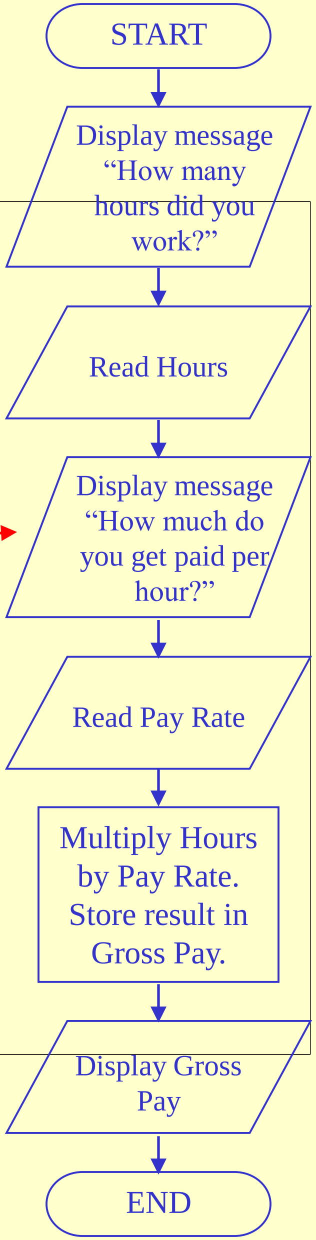
Pay Rate: ?

Gross Pay: ?

Stepping Through the Flowchart



Output Operation


START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

Read Pay Rate

Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

END

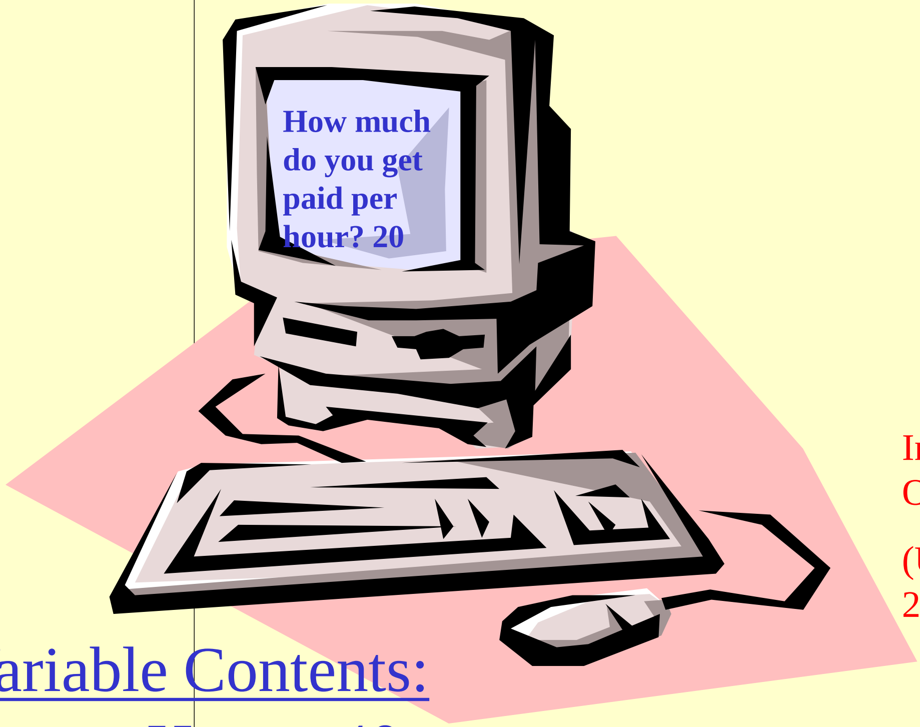
Variable Contents:

Hours: 40

Pay Rate: ?

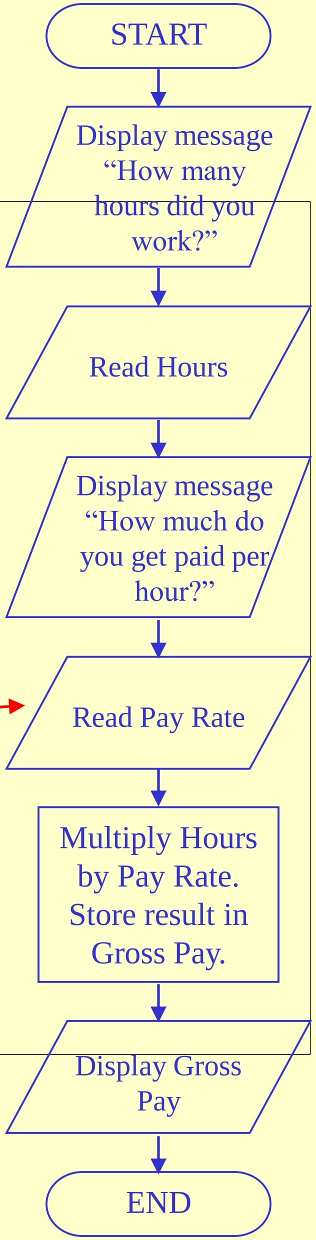
Gross Pay: ?

Stepping Through the Flowchart



Input Operation

(User types 20)


START
Display message "How many hours did you work?"
Read Hours
Display message "How much do you get paid per hour?"
Read Pay Rate
Multiply Hours by Pay Rate. Store result in Gross Pay.
Display Gross Pay
END

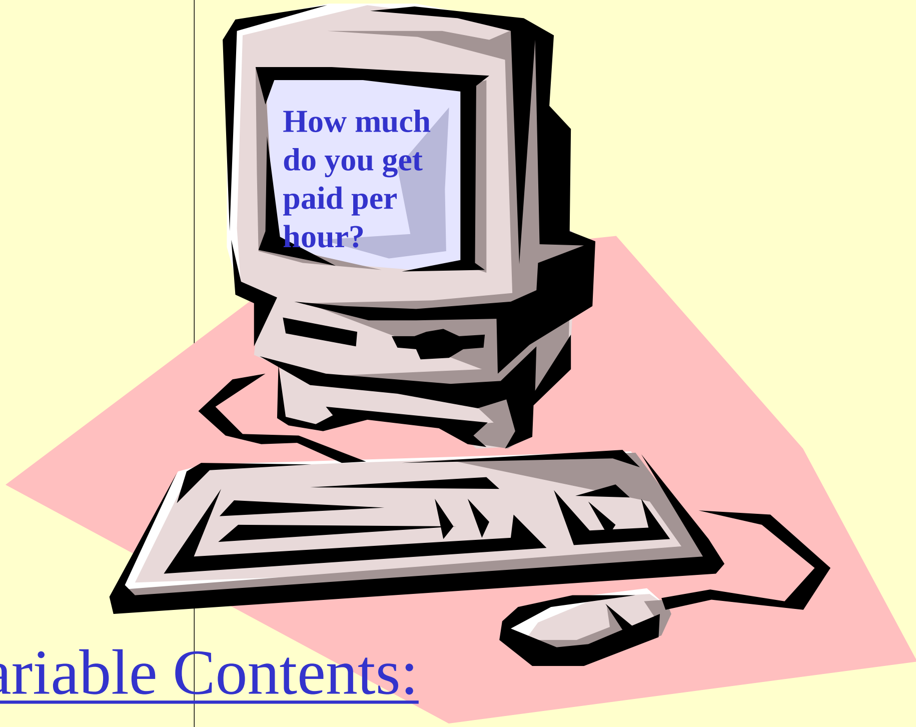
Variable Contents:

Hours: 40

Pay Rate: 20

Gross Pay: ?

Stepping Through the Flowchart



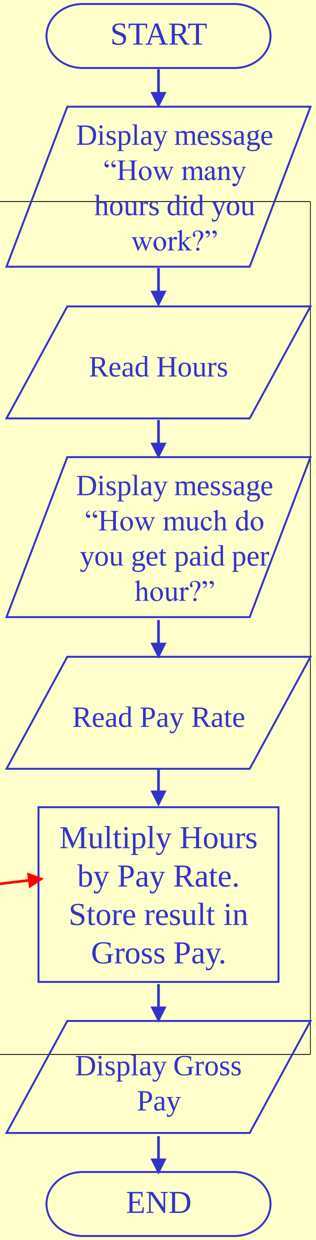
Variable Contents:

Hours: 40

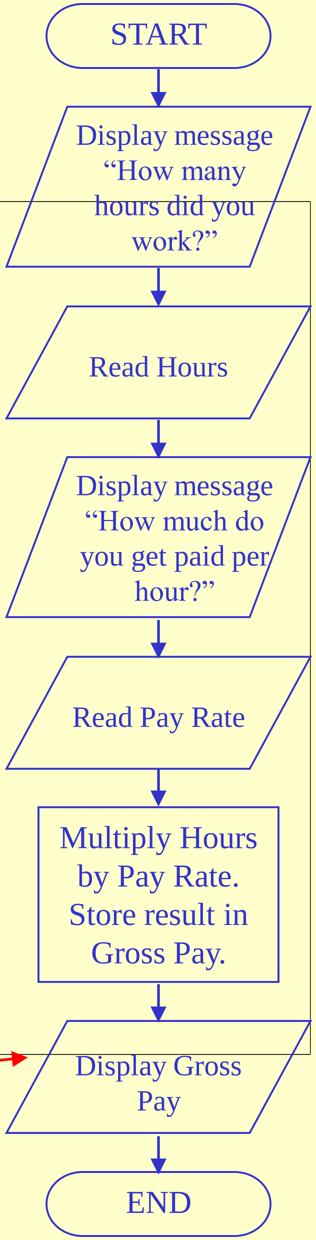
Pay Rate: 20

Gross Pay: 800

Process: The product of 40 times 20 is stored in Gross Pay


START
Display message “How many hours did you work?”
Read Hours
Display message “How much do you get paid per hour?”
Read Pay Rate
Multiply Hours by Pay Rate. Store result in Gross Pay.
Display Gross Pay
END

Stepping Through the Flowchart


START

Display message “How many hours did you work?”

Read Hours

Display message “How much do you get paid per hour?”

Read Pay Rate

Multiply Hours by Pay Rate. Store result in Gross Pay.

Display Gross Pay

END

Output Operation

Variable Contents:

Hours: 40

Pay Rate: 20

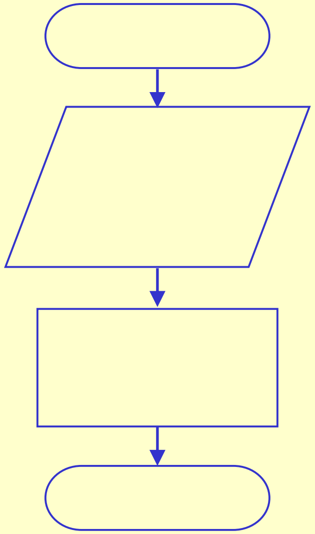
Gross Pay: 800

Four Flowchart Structures

- Sequence
- Decision
- Repetition
- Case

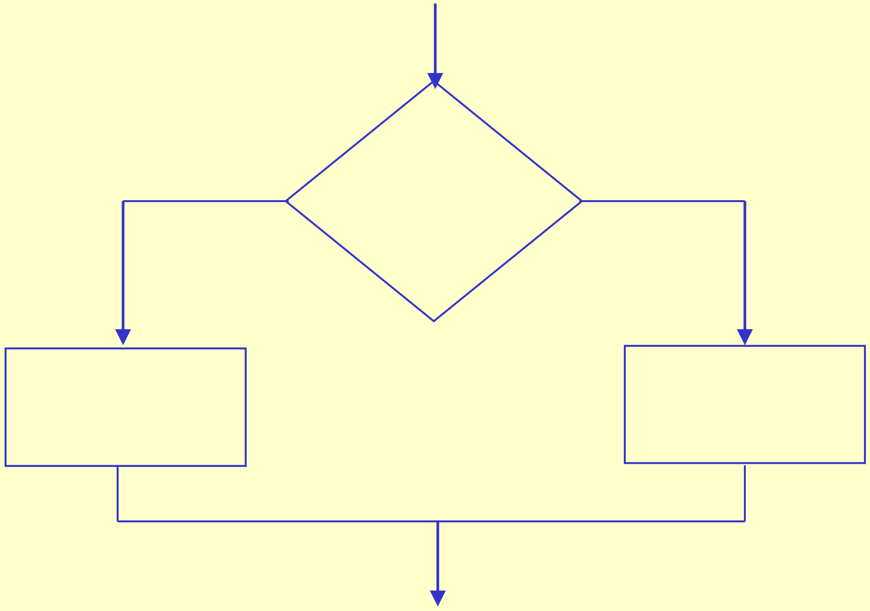
Sequence Structure

- a series of actions are performed in sequence
- The pay-calculating example was a sequence flowchart.



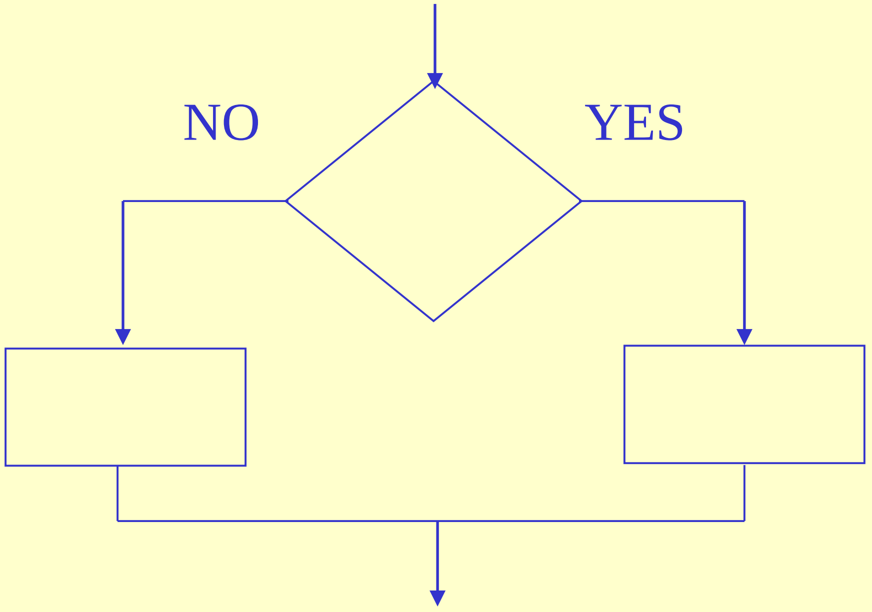
Decision Structure

- One of two possible actions is taken, depending on a condition.



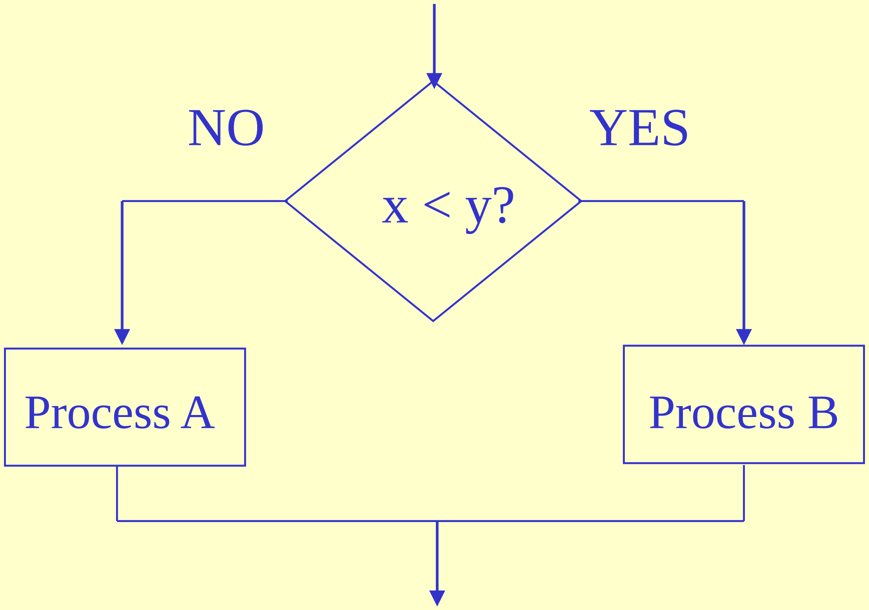
Decision Structure

- A new symbol, the diamond, indicates a yes/no question. If the answer to the question is yes, the flow follows one path. If the answer is no, the flow follows another path



Decision Structure

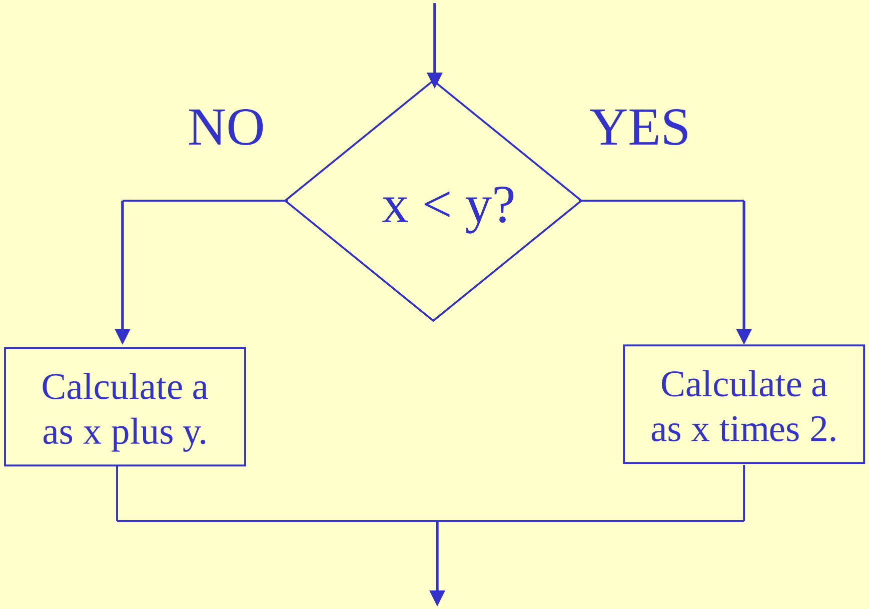
- In the flowchart segment below, the question “is x < y?” is asked. If the answer is no, then process A is performed. If the answer is yes, then process B is performed.



Decision Structure

- The flowchart segment below shows how a decision structure is expressed in C++ as an if/else statement.

Flowchart



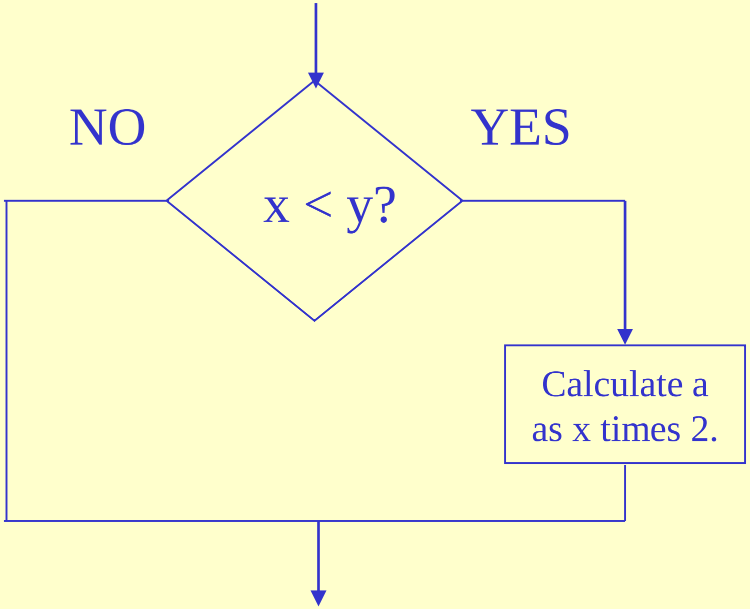
C++ Code

```
if (x < y)
        a = x * 2;
else
        a = x + y;
```

Decision Structure

- The flowchart segment below shows a decision structure with only one action to perform. It is expressed as an if statement in C++ code.

Flowchart

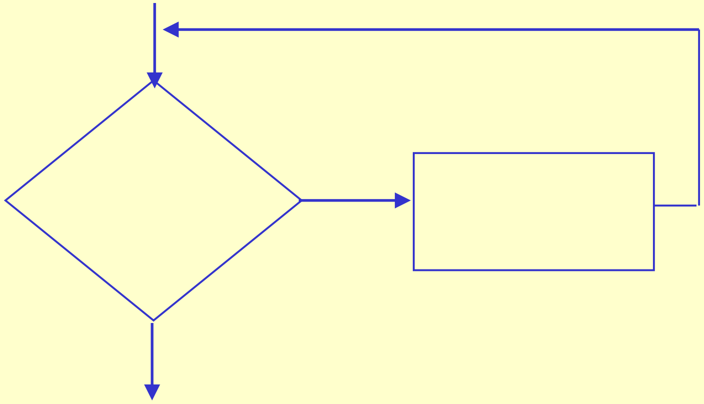


C++ Code

```
if (x < y)
     a = x * 2;
```

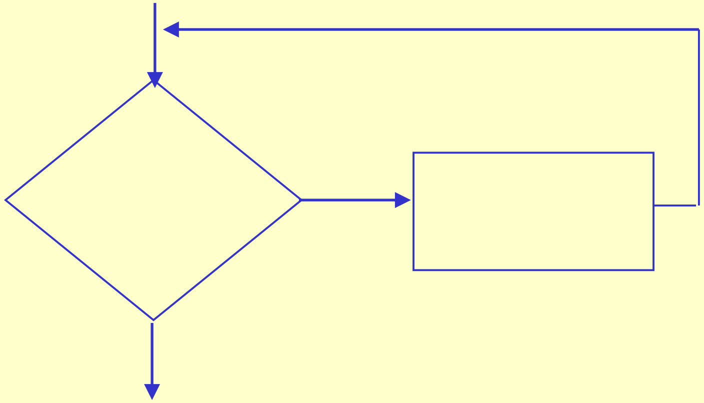
Repetition Structure

- A repetition structure represents part of the program that repeats. This type of structure is commonly known as a loop.



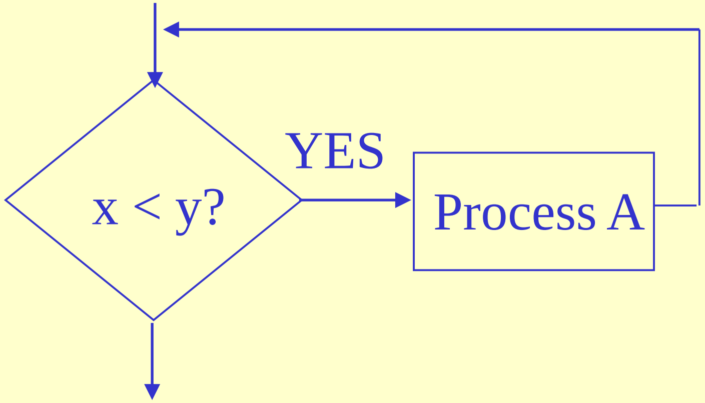
Repetition Structure

- Notice the use of the diamond symbol. A loop tests a condition, and if the condition exists, it performs an action. Then it tests the condition again. If the condition still exists, the action is repeated. This continues until the condition no longer exists.



Repetition Structure

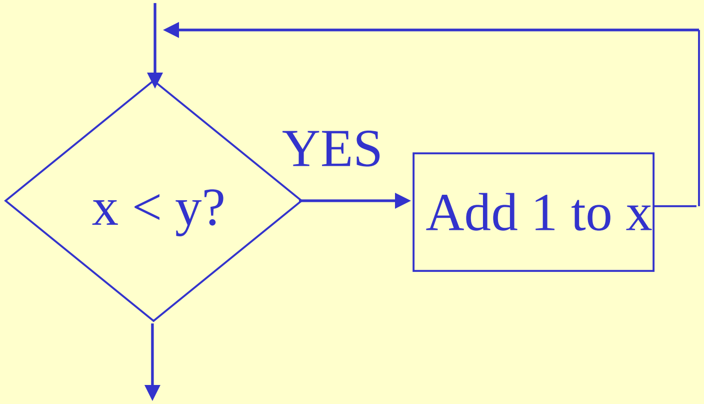
- In the flowchart segment, the question “is x < y?” is asked. If the answer is yes, then Process A is performed. The question “is x < y?” is asked again. Process A is repeated as long as x is less than y. When x is no longer less than y, the repetition stops and the structure is exited.



Repetition Structure

- The flowchart segment below shows a repetition structure expressed in C++ as a while loop.

Flowchart

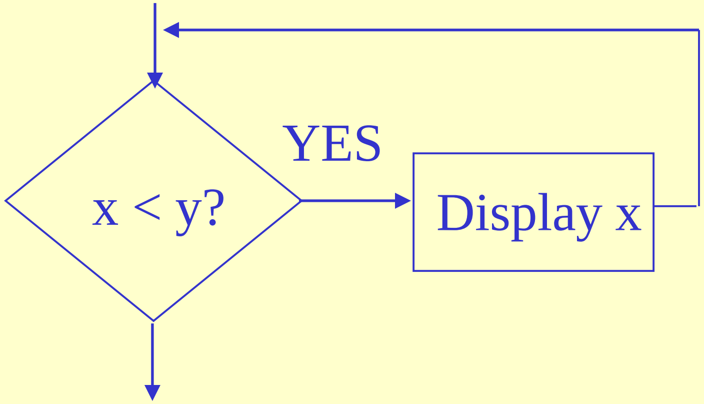


C++ Code

```
while (x < y)
    x++;
```

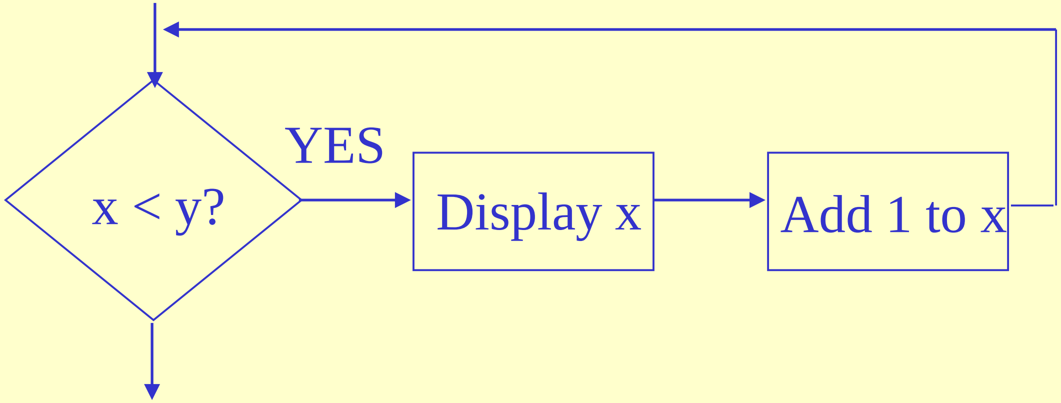
Controlling a Repetition Structure

- The action performed by a repetition structure must eventually cause the loop to terminate. Otherwise, an infinite loop is created.
- In this flowchart segment, x is never changed. Once the loop starts, it will never end.
- QUESTION: How can this flowchart be modified so it is no longer an infinite loop?



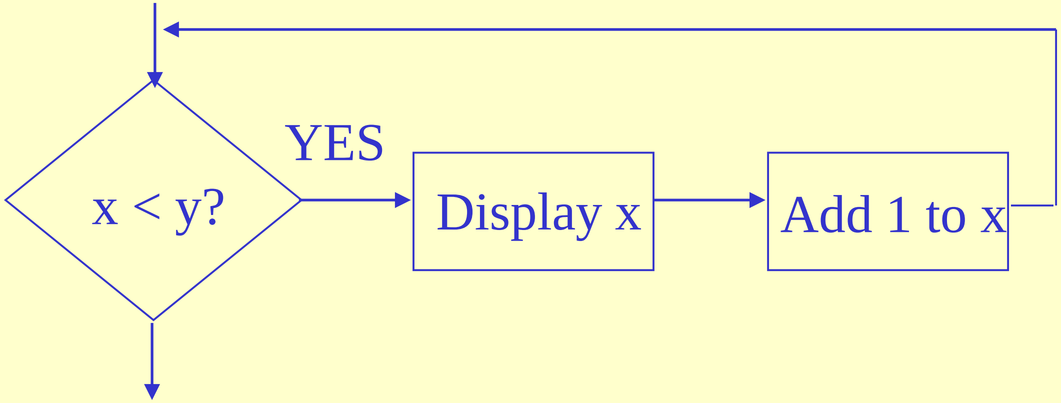
Controlling a Repetition Structure

- ANSWER: By adding an action within the repetition that changes the value of x.



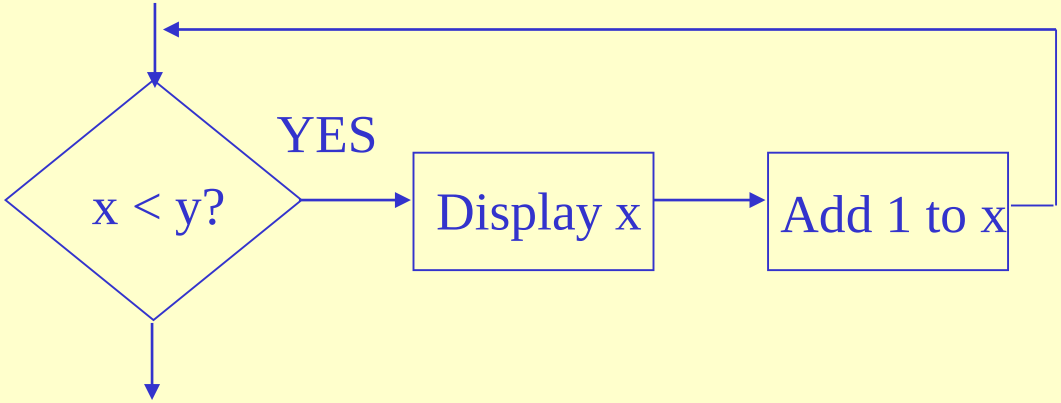
A Pre-Test Repetition Structure

- This type of structure is known as a pre-test repetition structure. The condition is tested *BEFORE* any actions are performed.



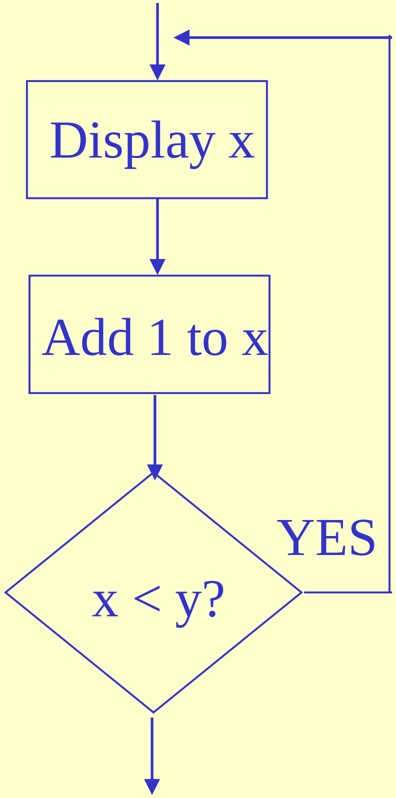
A Pre-Test Repetition Structure

- In a pre-test repetition structure, if the condition does not exist, the loop will never begin.



A Post-Test Repetition Structure

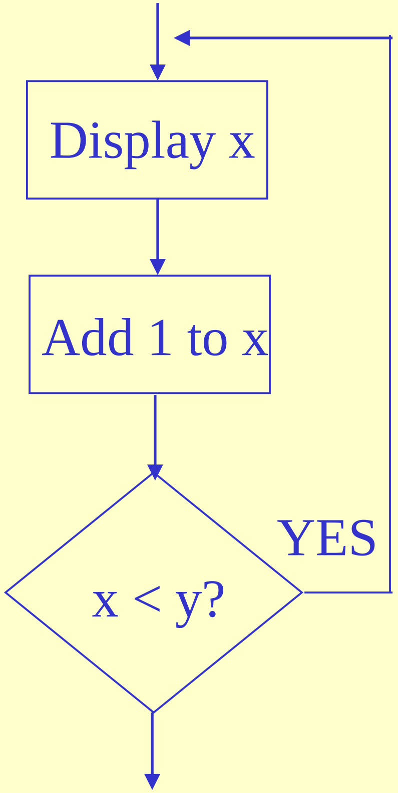
- This flowchart segment shows a post-test repetition structure.
- The condition is tested *AFTER* the actions are performed.
- A post-test repetition structure always performs its actions at least once.



A Post-Test Repetition Structure

- The flowchart segment below shows a post-test repetition structure expressed in C++ as a do-while loop.

Flowchart

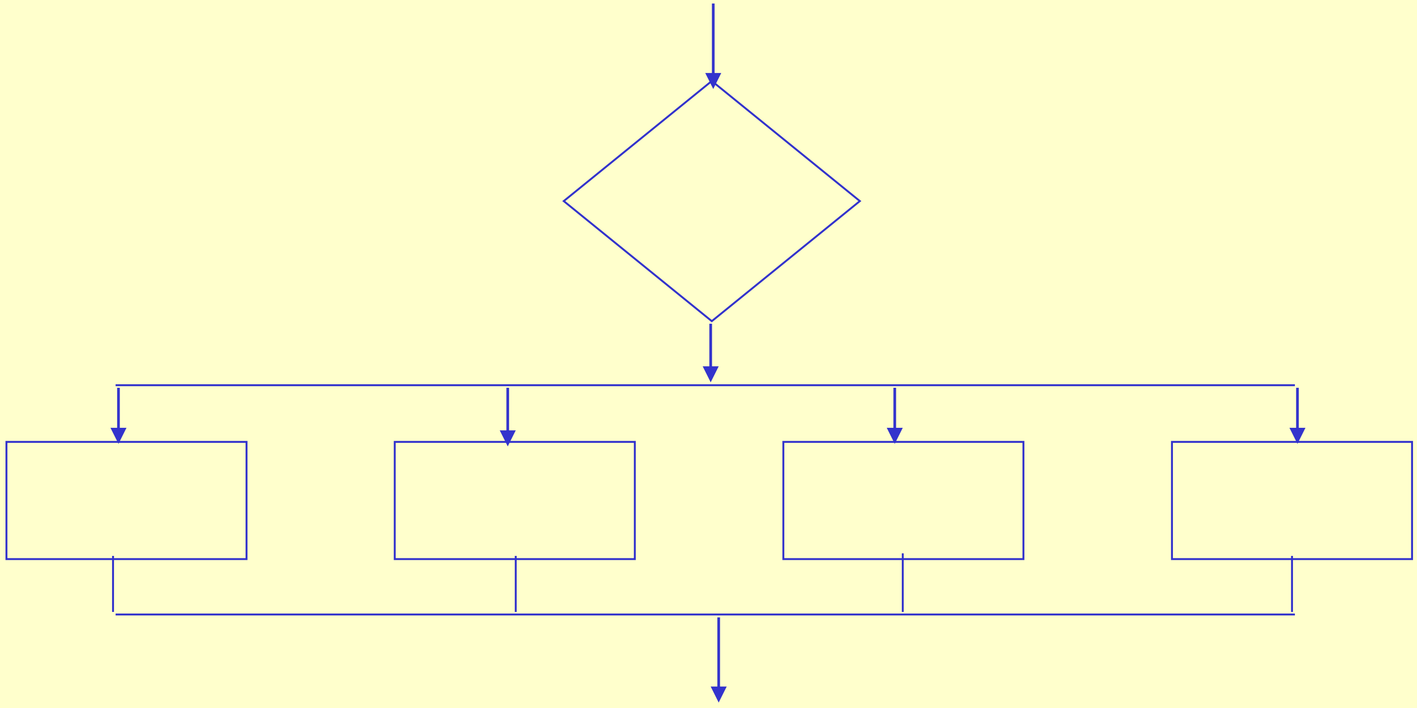


C++ Code

```
do
{
        cout << x << endl;
        x++;
} while (x < y);
```

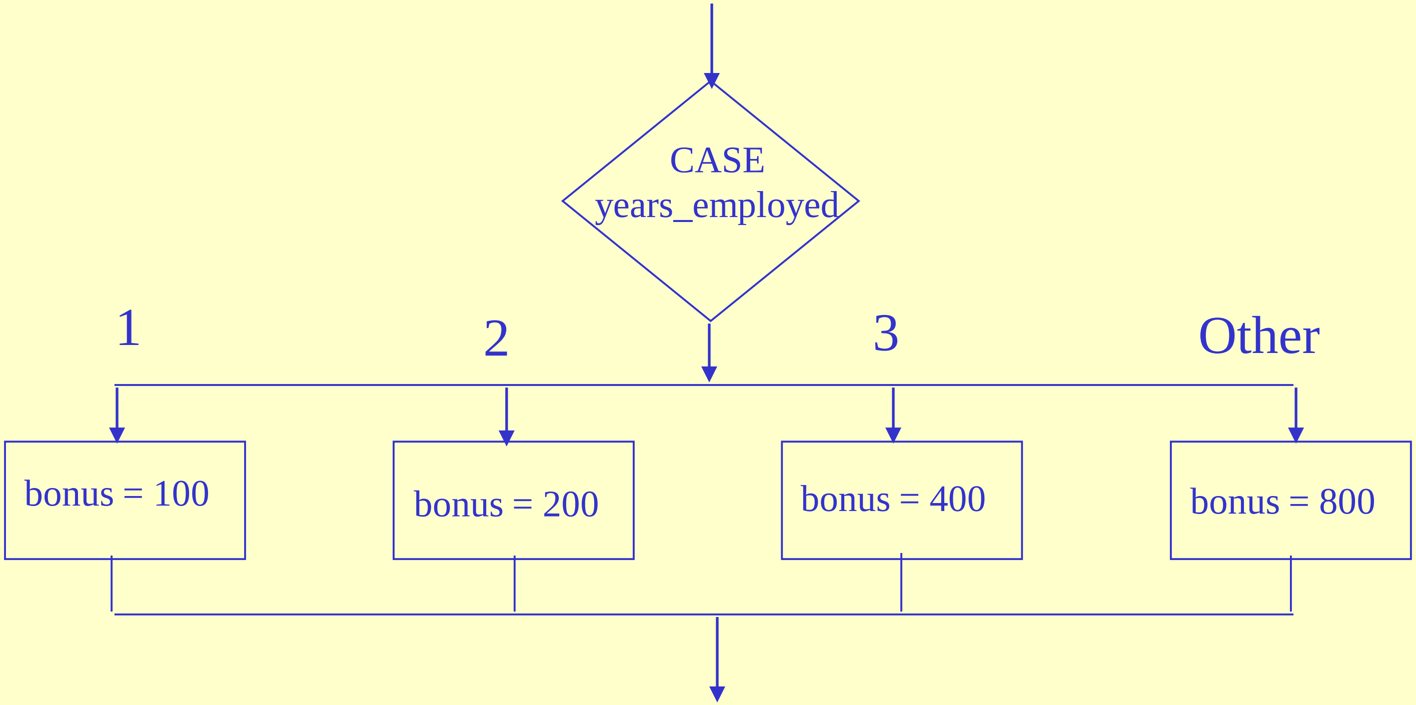
Case Structure

- One of several possible actions is taken, depending on the contents of a variable.

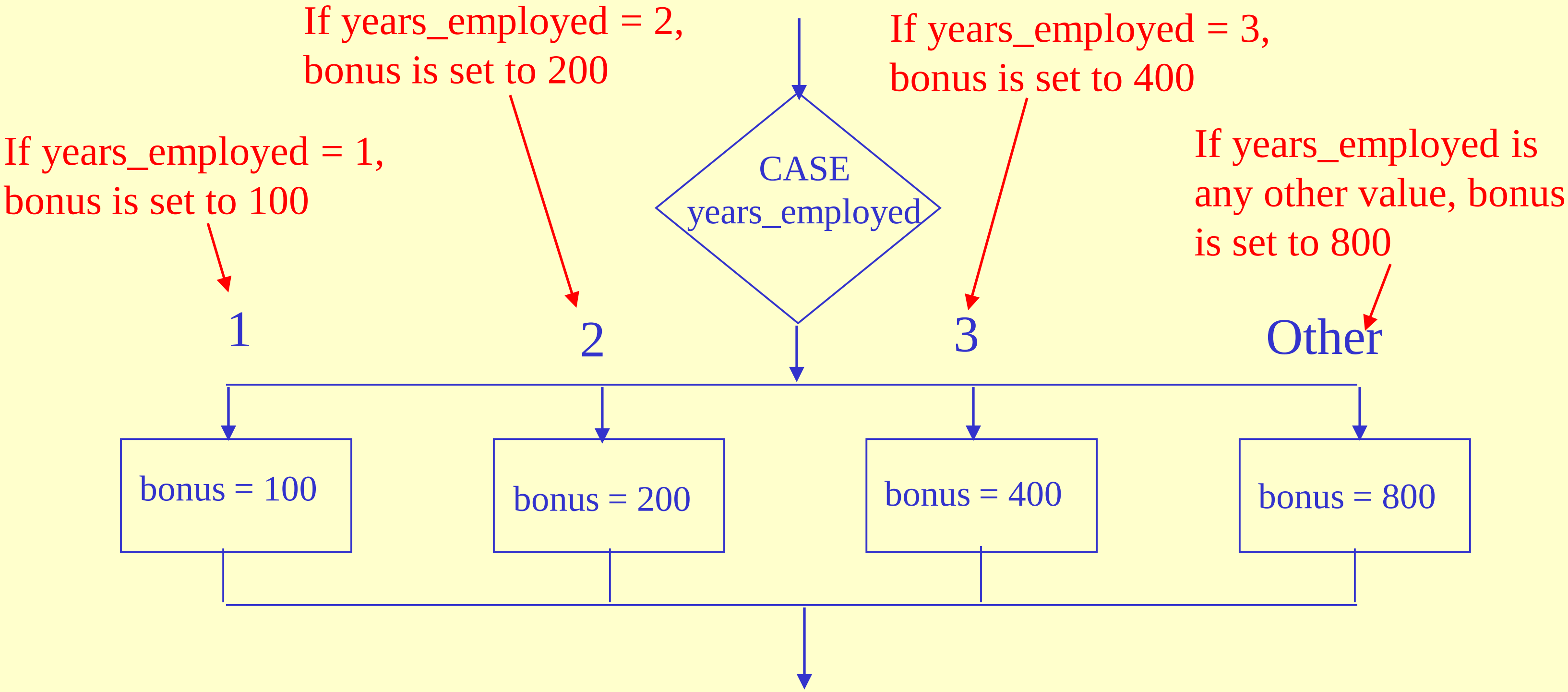


Case Structure

- The structure below indicates actions to perform depending on the value in `years_employed`.



Case Structure



If years_employed = 2,
bonus is set to 200

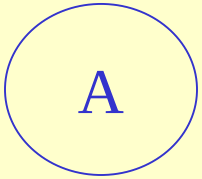
If years_employed = 3,
bonus is set to 400

If years_employed = 1,
bonus is set to 100

If years_employed is
any other value, bonus
is set to 800

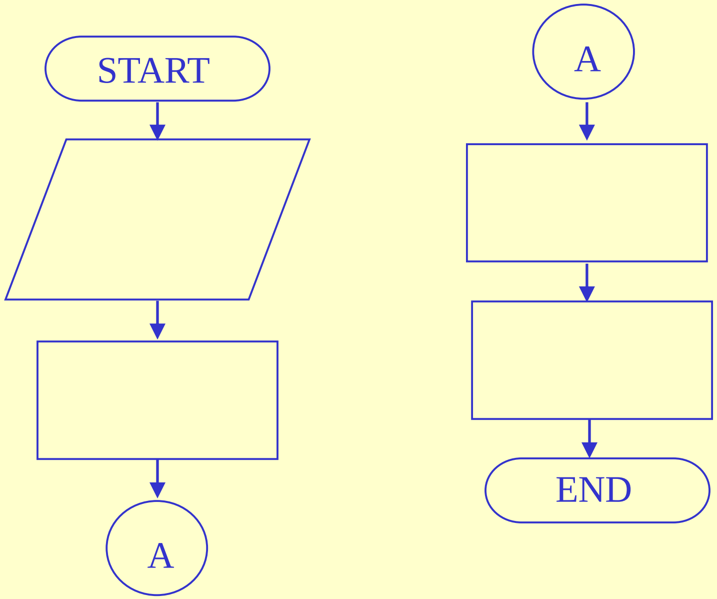
Connectors

- Sometimes a flowchart will not fit on one page.
- A connector (represented by a small circle) allows you to connect two flowchart segments.



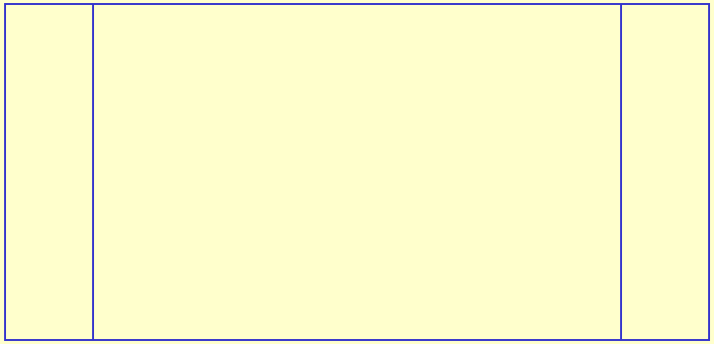
Connectors

- The “A” connector indicates that the second flowchart segment begins where the first segment ends.



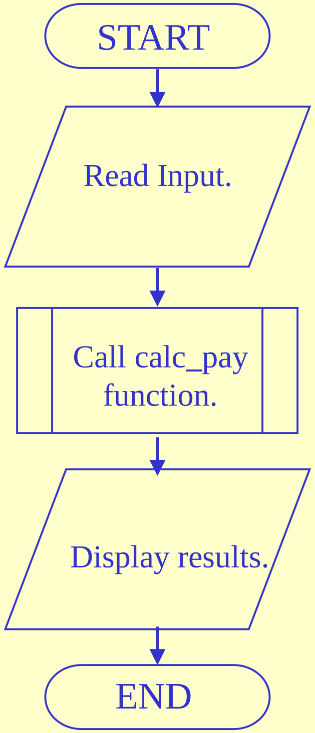
Modules

- A program module (such as a function in C++) is represented by a special symbol.



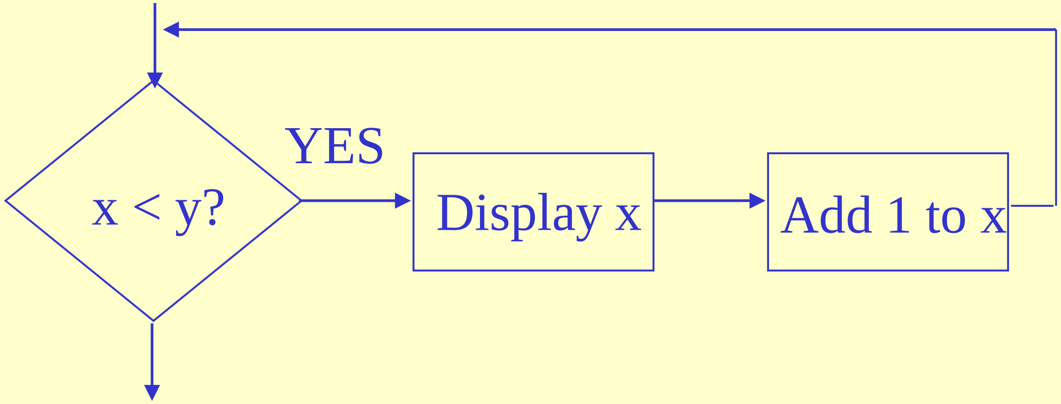
Modules

- The position of the module symbol indicates the point the module is executed.
- A separate flowchart can be constructed for the module.



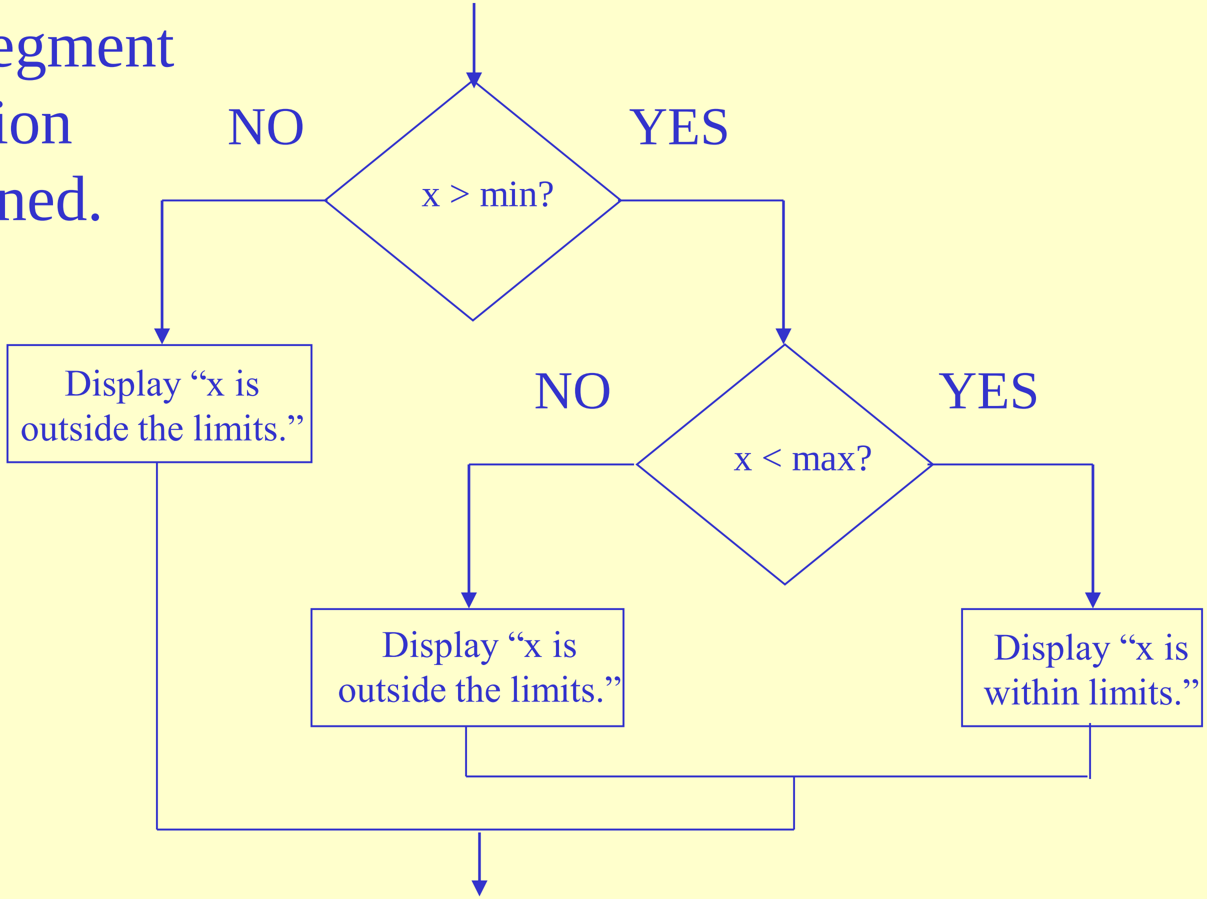
Combining Structures

- Structures are commonly combined to create more complex algorithms.
- The flowchart segment below combines a decision structure with a sequence structure.



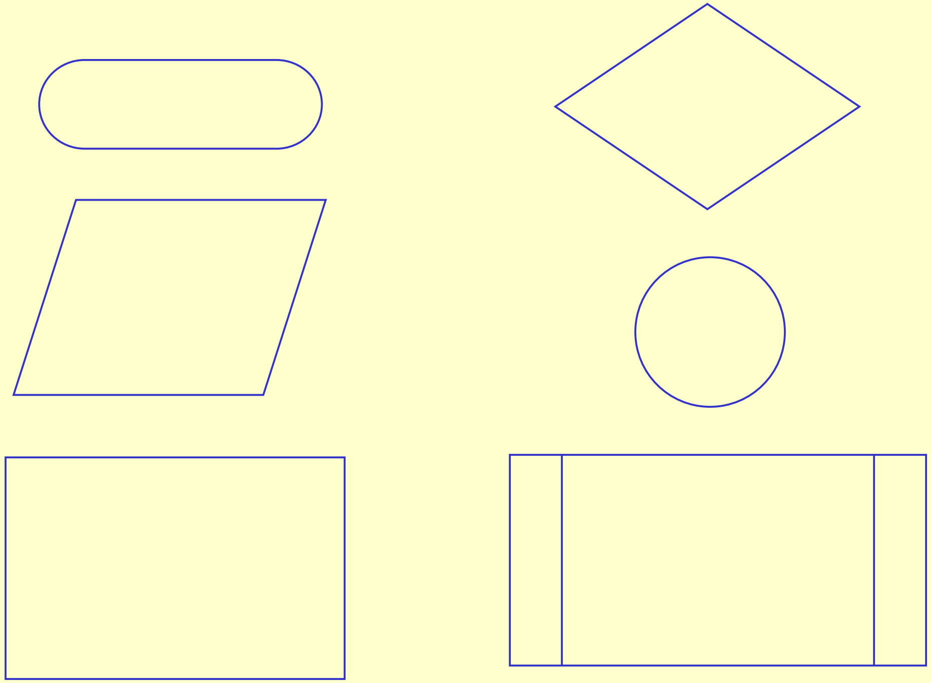
Combining Structures

- This flowchart segment shows two decision structures combined.


NO
x > min?
YES
Display "x is outside the limits."
NO
x < max?
YES
Display "x is outside the limits."
Display "x is within limits."

Review

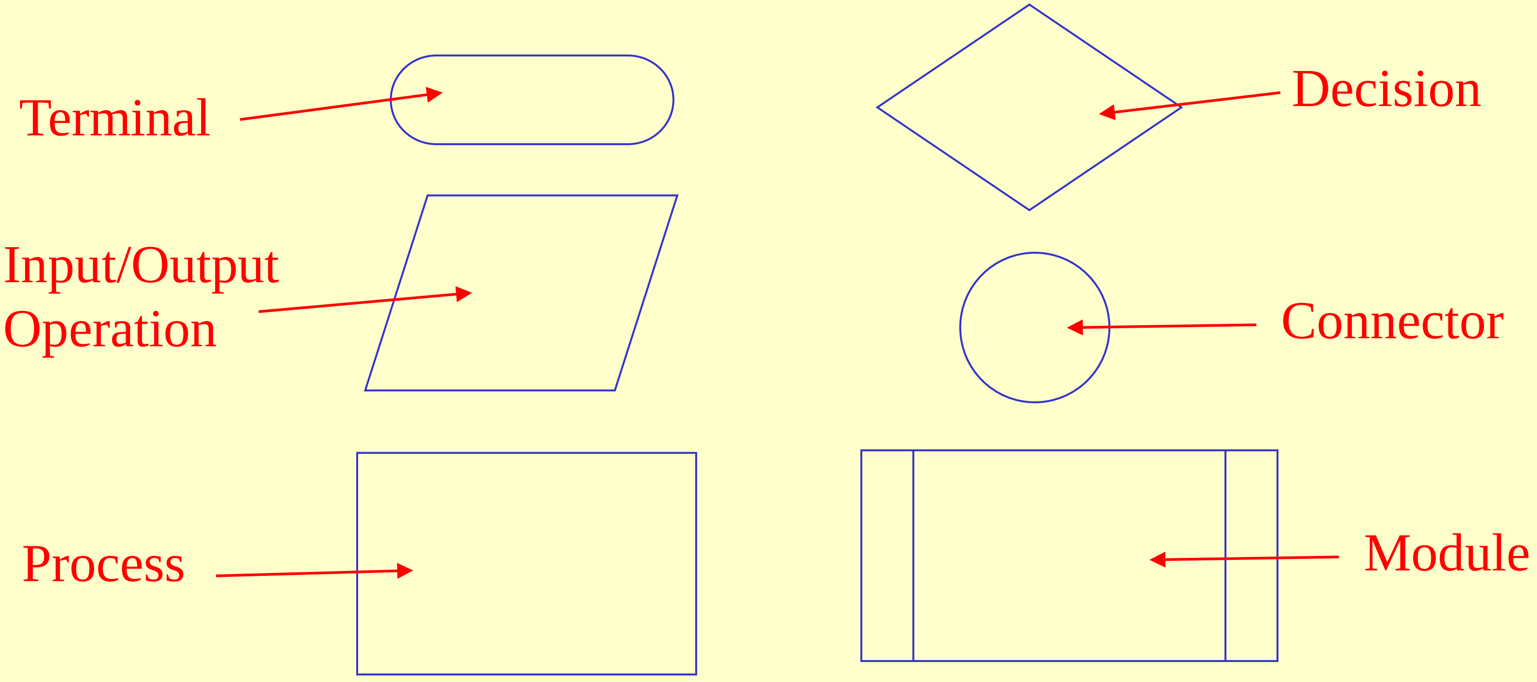
- What do each of the following symbols represent?



(Answer on next slide)

Answer

- What do each of the following symbols represent?



Review

- Name the four flowchart structures.

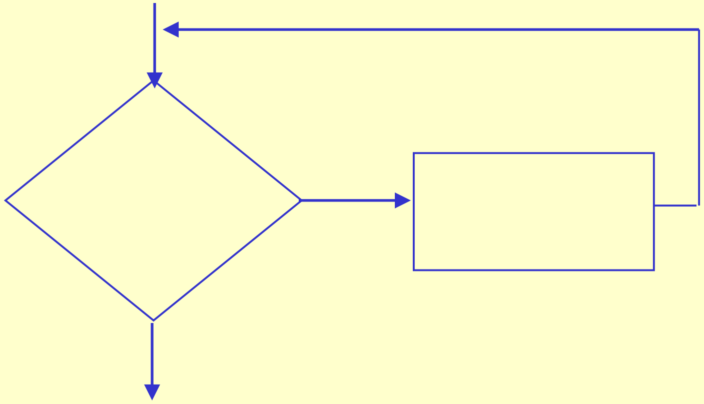
(Answer on next slide)

Answer

- Sequence
- Decision
- Repetition
- Case

Review

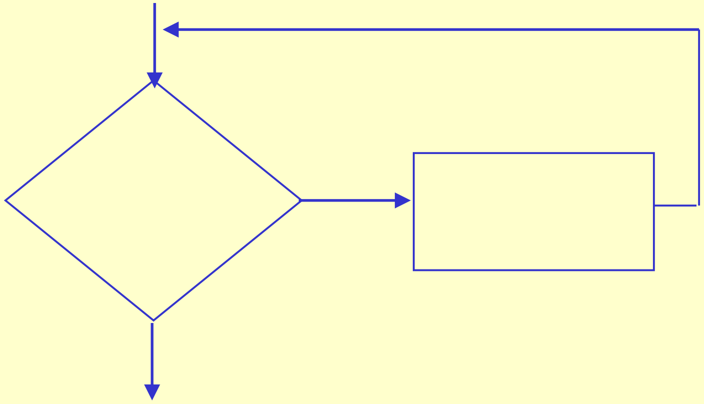
- What type of structure is this?



(Answer on next slide)

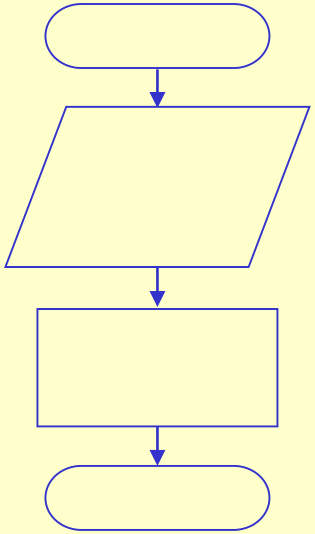
Answer

- Repetition



Review

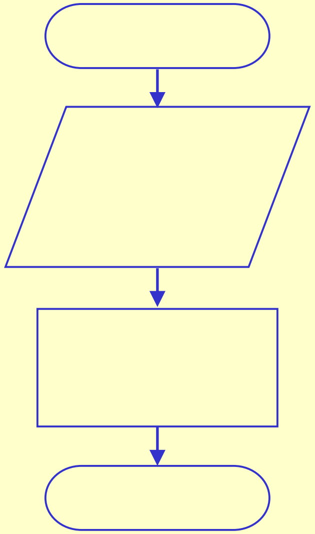
- What type of structure is this?



(Answer on next slide)

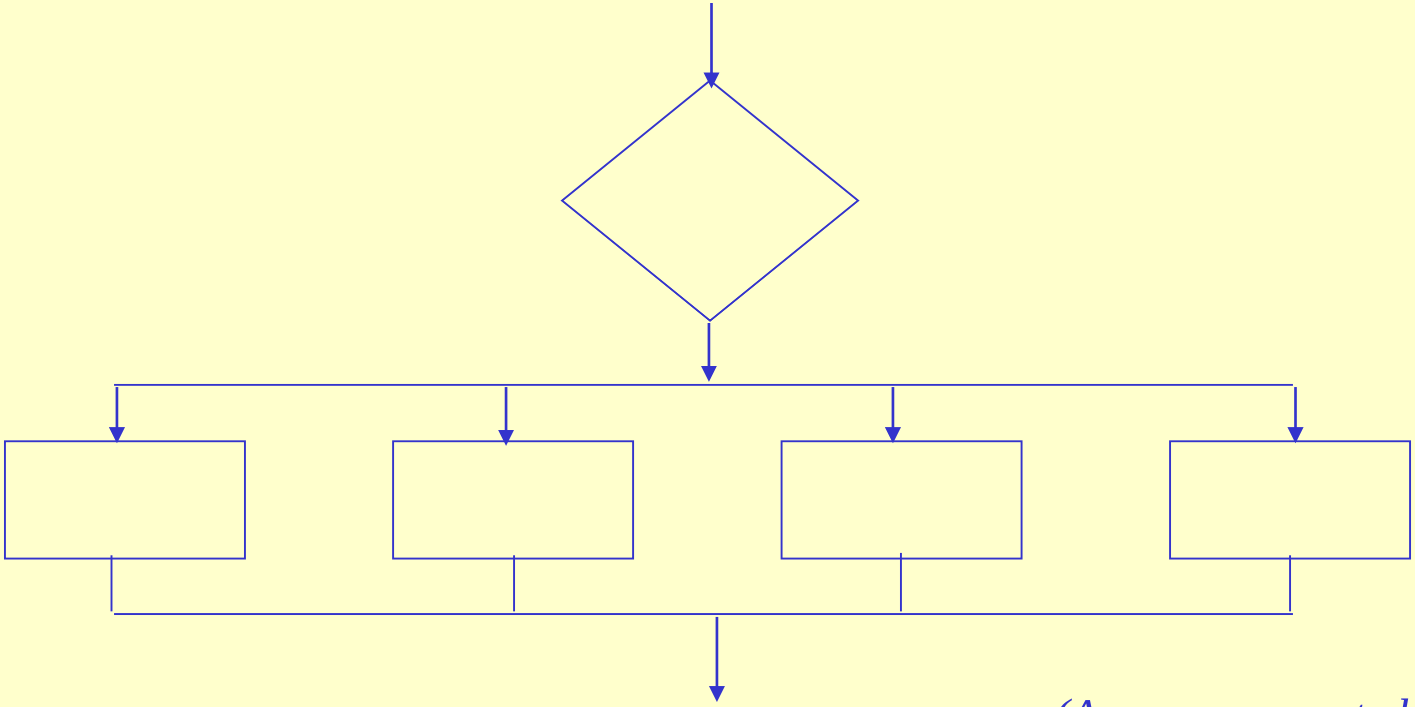
Answer

- Sequence



Review

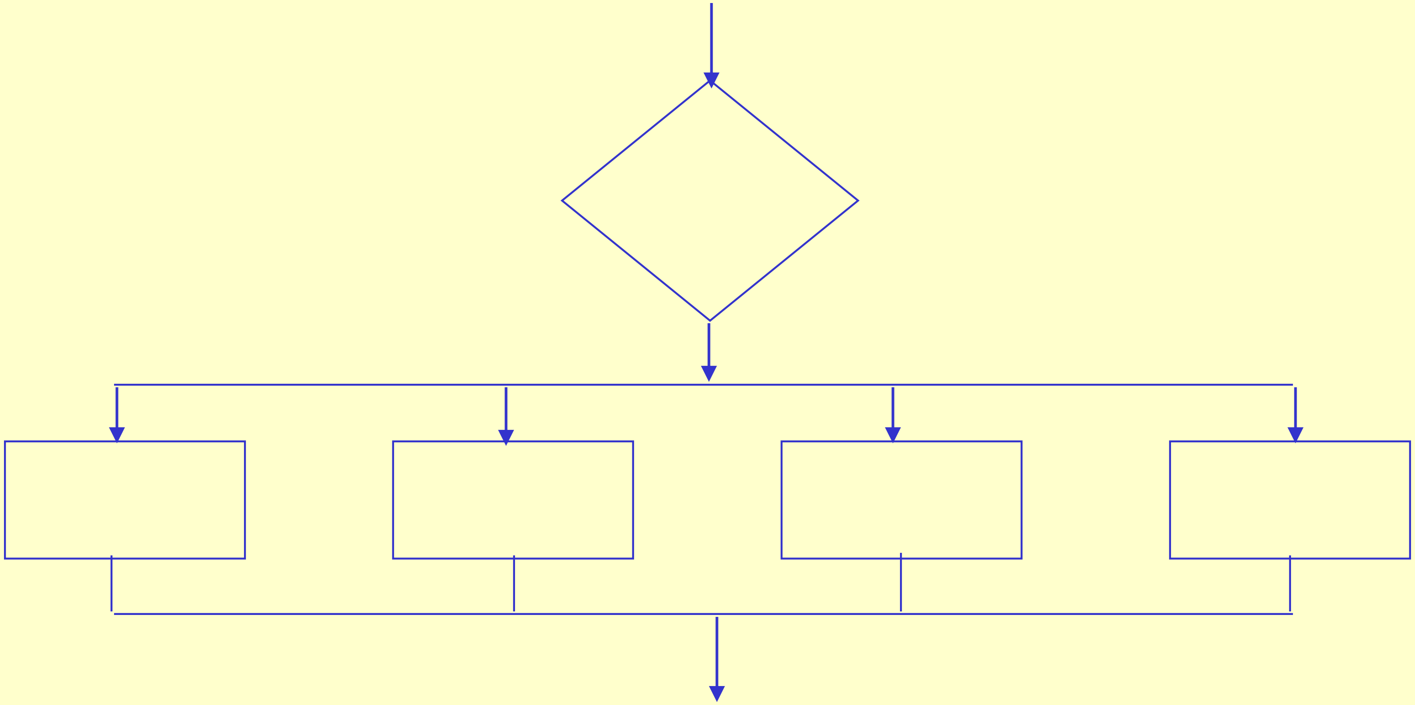
- What type of structure is this?



(Answer on next slide)

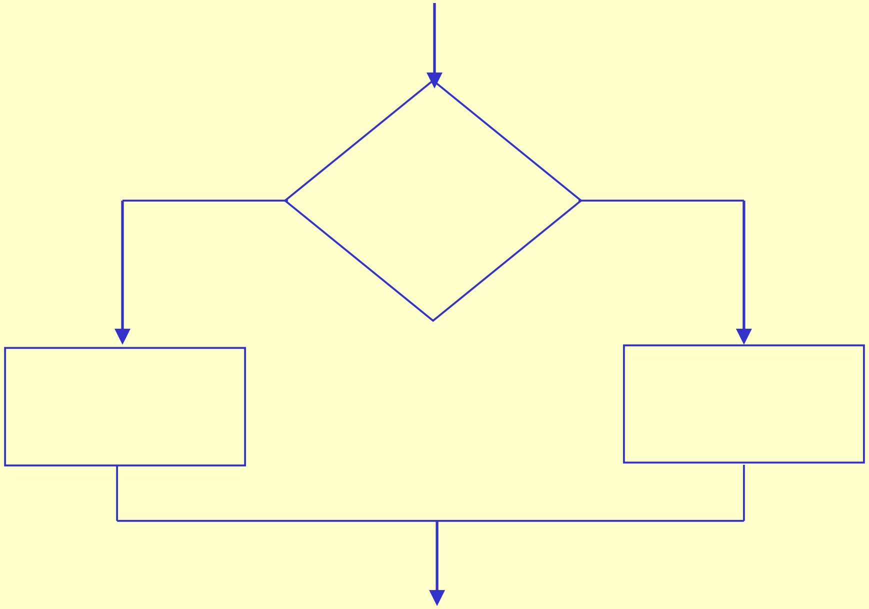
Answer

- Case



Review

- What type of structure is this?



(Answer on next slide)

Answer

- Decision