9

DepED
DEPARTMENT OF EDUCATION

SELF-LEARNING PACKAGE I

ICT 9

Quarter 1 | Week 3

Control Structures

Learning Competency:

Describe how control structures is used in programming

SSP_TLE-CT8AP-IId-m-3.1

MARICAR R. PORNEL, *Oton NHS*
ZALDY M. TONDO, *Division Science Coordinator*
WRITERS

Ready to Launch!

A program is a sequence of instructions. In the ordinary "flow of control", the computer executes the instructions in the sequence in which they appear, one after the other.

In this lesson, we will discuss how program instruction works, how do we read program code line by line (form top to bottom, and from left to right). At some point, the program may reach a situation where it needs to make a decision such as jump to a different part of the program or re-run a certain piece again. These decisions that affect the flow of the program's code are known as a **Control Structures**.

Aim at the Target!

At the end of this module you are expected to:

1. Identify each of the control structures.
2. Explain the flow of instructions for each of the control structures.

Try This!

Direction. Unscramble the following words with the help of the given clue.

WORD	CLUE
POLO	The purpose is to repeat a statement a certain number of times or while a condition is fulfilled.
TCEELSNIO	It is used to execute one or more statements if a condition is met.
UENEESQC	Set of ordered instructions executed by a computer one after another.
OOLRTCL ERUTCURTS	Commands that enable a program to take decisions, following one path or another
ROF POLO	Allows a statement to be executed a specified a number of times.

Keep This in Mind!

As mentioned earlier, we read program instructions line by line starting from top to bottom, and from left to right. This is the same way the ordered instructions are executed by a computer. An example of it can be seen in figures A and B below.

Activity 1. Reading program instruction

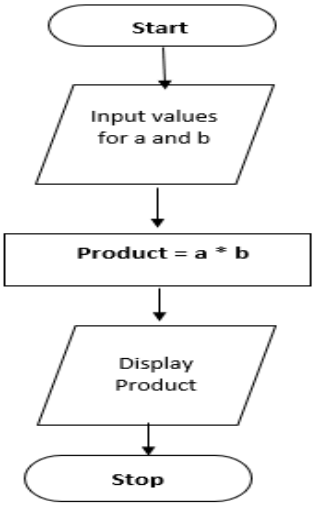


Figure A

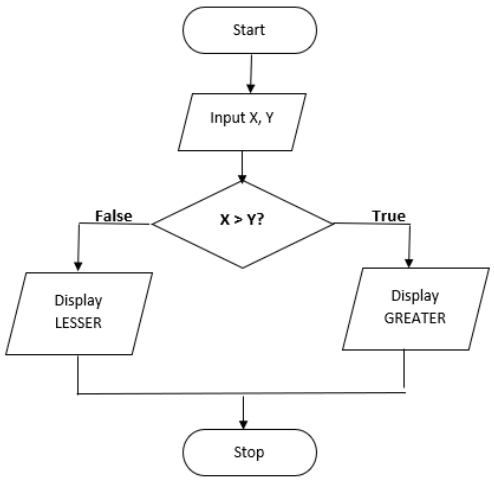


Figure B

Analysis:

1. Enumerate how you read the program instructions (from top to bottom) in figure A.
2. Enumerate how you read the program instructions (from top to bottom and left to right) in figure B.

Abstraction and Generalization

What is control structure?

Control Structure can be considered as the building blocks of computer programs. They are commands that enable a program to "take decisions", following one path or another. A program is usually not limited to a linear sequence of instructions since during its process it may bifurcate, repeat code or bypass sections. Control Structures are the blocks that analyze variables and choose directions in which to go based on given parameters.

Type of Control Structure

1. **Sequence**

The sequence structure directs the computer to process program instructions one after the another, in the order listed in the program.

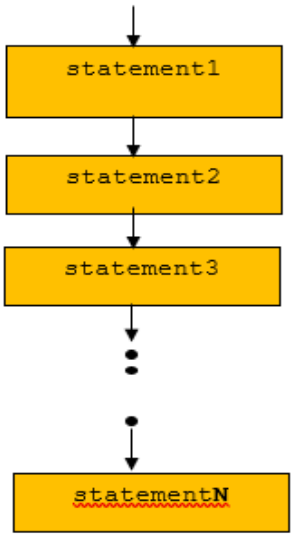


Fig. 1 Sequence structure

Example:

Supposed we want to compute for the sum of two numbers. In order to do that, we need to get the values of the two number, add the numbers, and display its sum. The flowchart below consists of different steps following the sequence of instructions in order to get the sum of the two numbers.

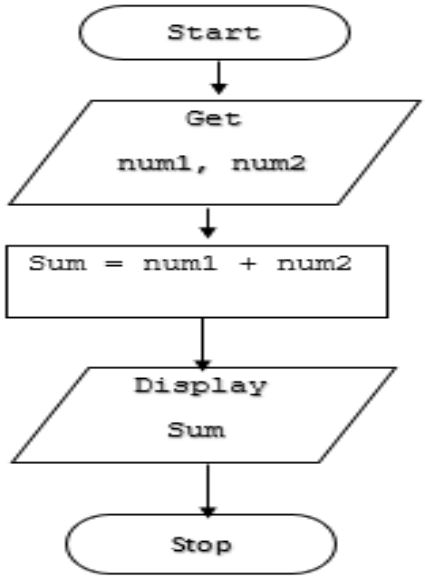


Fig. 2 Sequence Example Flowchart:

2. **Selection**

The Selection structure allows the computer to decide between two or more different courses of action by testing conditions that occur as the program is running.

It also called **decision** structure.

There are two types of Selection structures:

A. Binary selection. There are two possible choices to choose. Binary uses IF… ELSE... ENDIF. If the condition is met(true) then one path is taken, otherwise (false) the other path is taken.

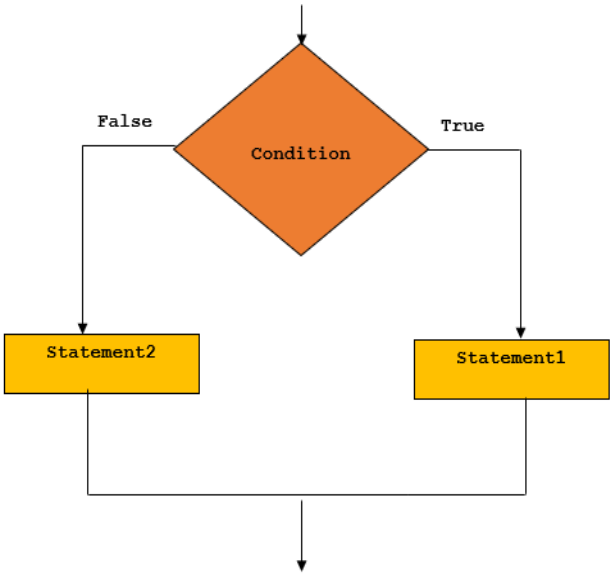


Fig. 3 Binary Selection structure

Example:

Supposed you are task to compute for the salary of employees. The computation will be based on their hours worked. A condition is set to determine whether it will compute for the regular pay or overtime pay.

The solution is illustrated in the example flowchart. Here, we can see that the path to be chosen will depend on the input of the employees number of hours worked. So, the action taken is dependent on the number of hours spent working. If greater than 40, then overtime pay is computed if not, it will compute for the regular pay. The last task is to print the paycheck of the employee.

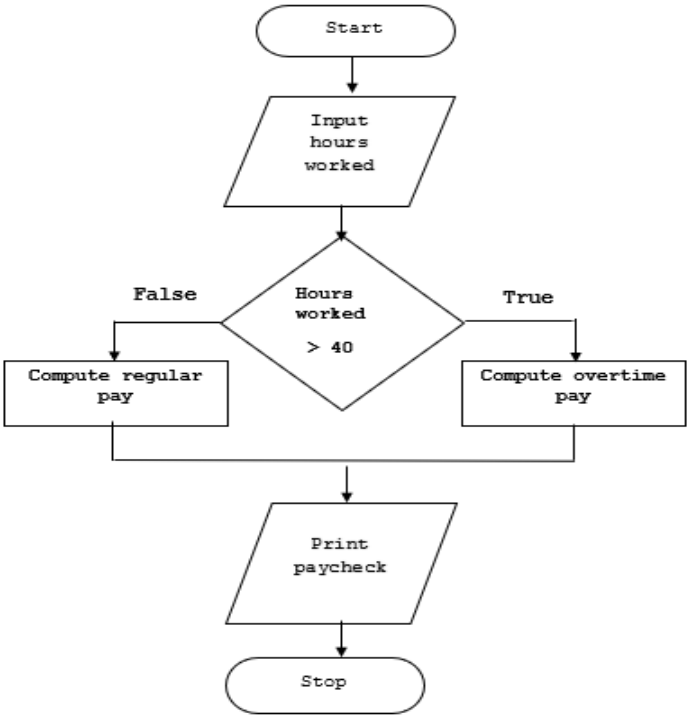


Fig. 4 Binary Selection Example Flowchart:

B. **Switch Case selection**

- Case selection is where there is more than one possible choices to choose when trying to solve the problem. Only one process can be carried out.
- The condition is checked and if the first choice is true then it is carried out. If the choice is false , then the second will be checked. If it is true then it is carried out. If no choice is found to be true then the otherwise choice will be carried out.

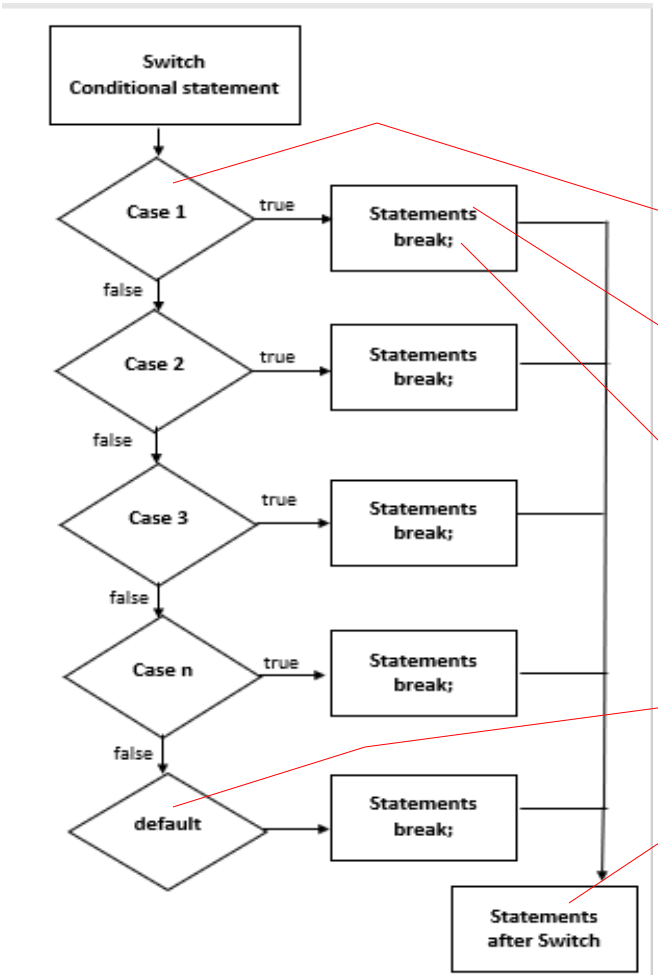


Fig. 5 Switch Case structure

`Switch` statement actually gathers many statements of several cases.

- the **`case`** labels denote the specific statement from which the execution of this group statements begins.

- all statements till the end of the group are executed sequentially.

To separate the cases, **`break statement`** is used.

- **`break`** breaks the sequential execution of the statements and immediately jumps to the end of the switch statement.

`Default` case is optional. It has no associated value. If the default case is present, control will transfer to it if no other case value matched.

If there is no default case, and no other vale matches, control falls through to the statement after the switch

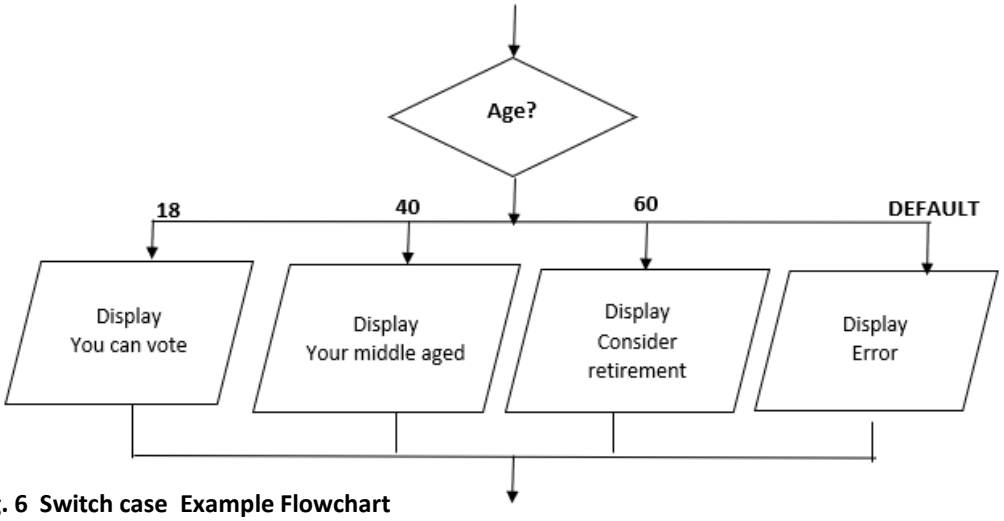


Fig. 6 Switch case Example Flowchart

In this example, The value in the variable **age** is compared to the first "**case**", which is the value 18 (also called the listed value) using an equality comparison or is "age equal to 18". If it is true, the message is assigned the value "You can vote." and the next line of code (the break) is done (which jumps us to the end of the control structure). If it is false, it moves on to the next case for comparison. The last item is referred to as the default. If the age is not equal to 18, 40 or 60 you get the default message " Error".

3. **Repetition**

The Repetition structure directs the computer to repeat one or more instructions until some condition is met.

- It is also called **loop** or **iteration**.
- The loop is created to return the program to where the repetition has started as long as it takes until the condition is met.

Types of loop:

1. While
2. Do-while
3. For loop

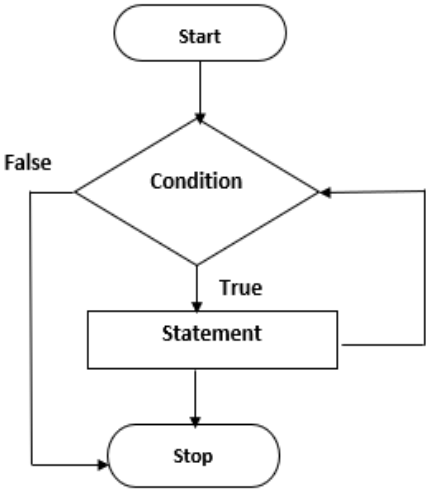


Fig. 7 While loop structure

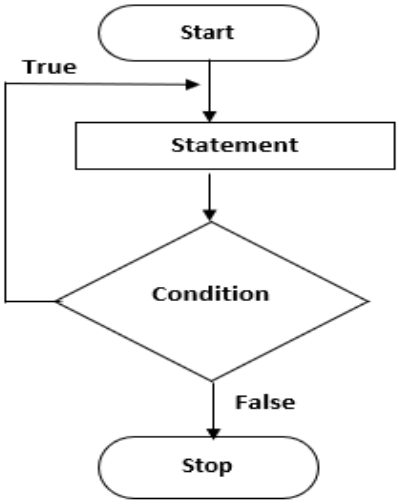


Fig. 8 Do-While loop structure

While and Do-While Loop Comparison

While Loop	Do-While Loop
The while loop evaluations the condition first and then execute the statements.	The do-while loop executes the statements first before evaluating the condition.
The condition is specified at the beginning of the loop.	The condition isn't specified until after the body of the loop.
The body is executed only if a certain condition is met and it terminates when the condition is false.	The body is always executed at least once, regardless of whether the condition is met.

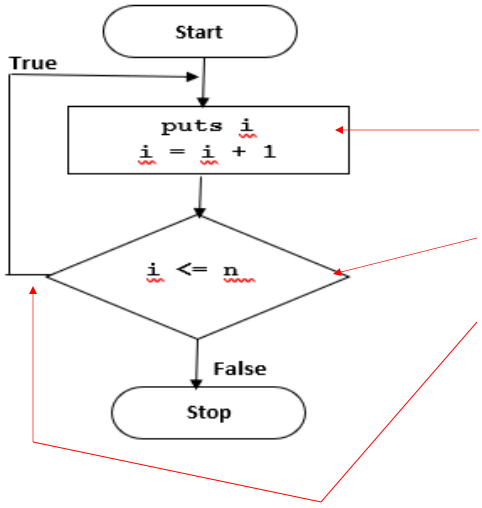


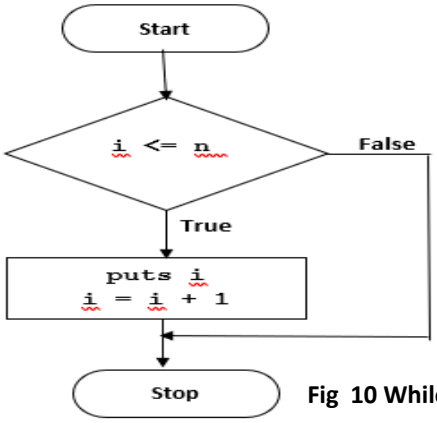
Fig 9 Do-While loop structure Example Flowchart

In this example, supposed we initialized the values of `i and n;`

```
i = 0
n = 5
```

- When we perform the this statement , the value of **`i`** is now equals to **`1 (0 + 1 = 1).`**
- When this condition (1 <= 5) is tested the result is **true**.
- So the control flow of instruction goes back to the statement and perform again the operation **`1 + 1,`** **and so the value `of i now is 2.`**
- The condition **`(2 <= 5)`** is tested again, giving the result of **True**
- The control flow of instruction goes back again to the statement and perform again the operation **`2 + 1,`** **and so the value `of i now is 3.`**

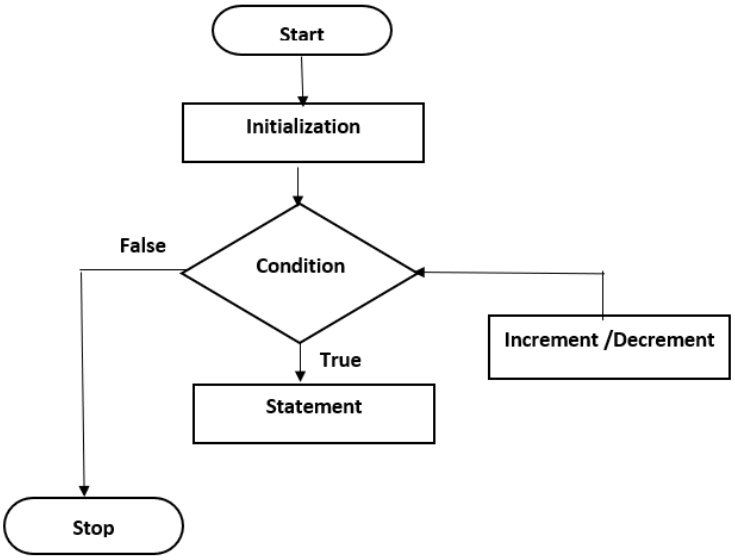
So the loop goes on and on until the condition becomes false, that is when **`i`** is equals to **6.**



If we take the same sample initialization values of **`i`** and **`n`** in the above example, the same thing will happen using While loop, except that in the flow of instruction it will begin by the testing the condition first then followed by executing the statement.

Fig 10 While loop structure Example Flowchart

For Loop allows a statement to be executed a specified a number of times.



The for loop begins with a loop control variable assigned a specific initial value. This control variable in then incremented(or decremented) by a specified amount each time around the loop until a specified terminating value is reached at which time the statement following the loop is then executed.

Fig. 11 For Loop Structure

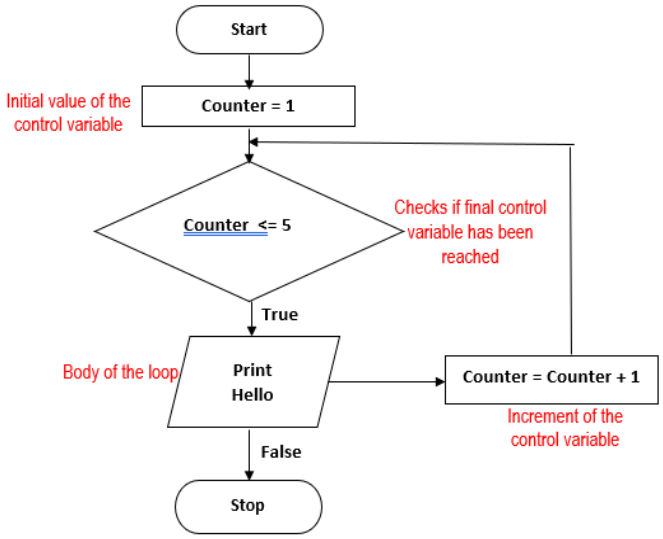


Fig 12 For loop structure Example Flowchart

In the example, counter is the control variable and **`counter = 1`** is the first statement which initializes counter to 1.

The second expression, **`counter <= 5`** is the loop continuation condition which repeats the loop till the value of counter exceeds 5.

Finally, the third expression, **`counter = counter + 1`** increases the value of counter by 1.

So the above piece of code will print 'hello' five times.

Application.

Direction. Identify the name of each control structures as shown by the following figures below.

Activity 2. Naming a Control Structure

Figure A

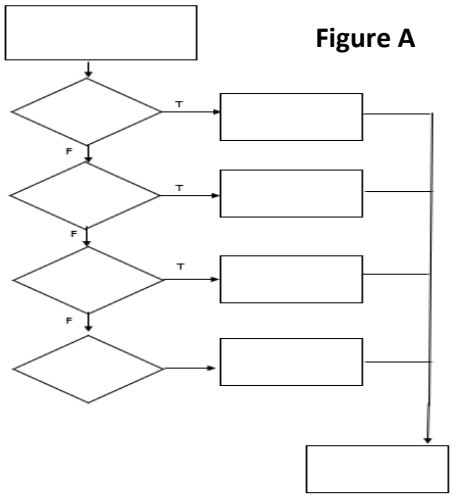


Figure B

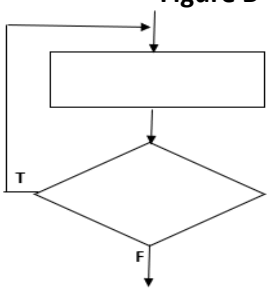


Figure C

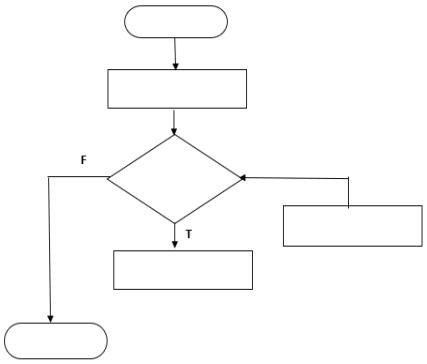


Figure D

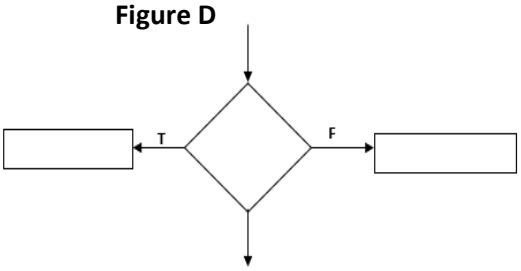


Figure E

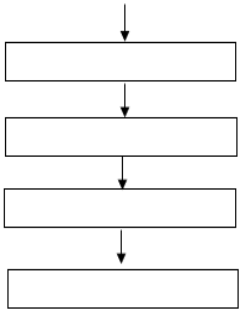
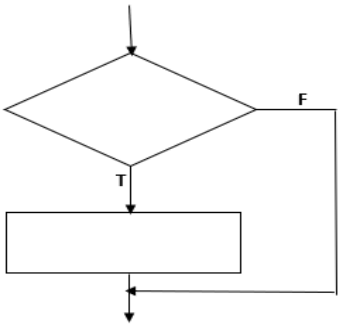


Figure F



Reflect

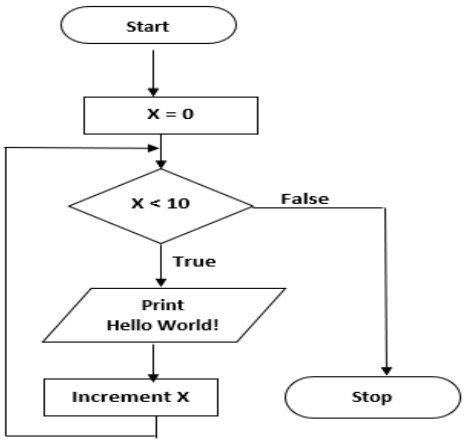
Complete the statements below.

I understand _____

I don't understand _____

I need more information about _____

Reinforcement & Enrichment

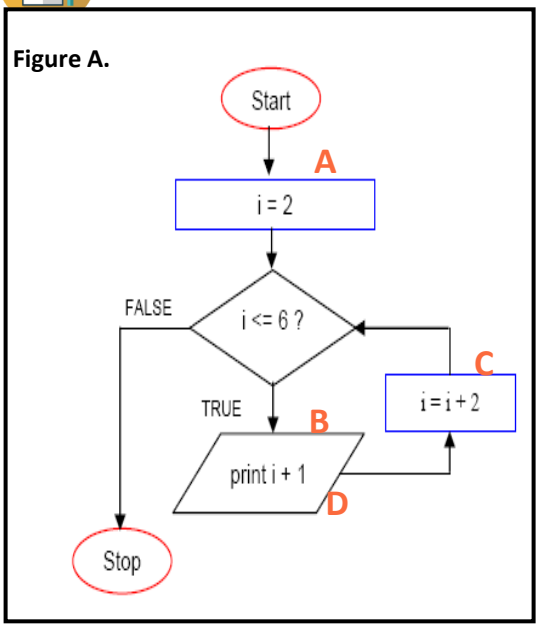


Direction. Answer the question below.

Explain the flow of instructions as shown in the flowchart.

Assess Your Learning

Figure A.

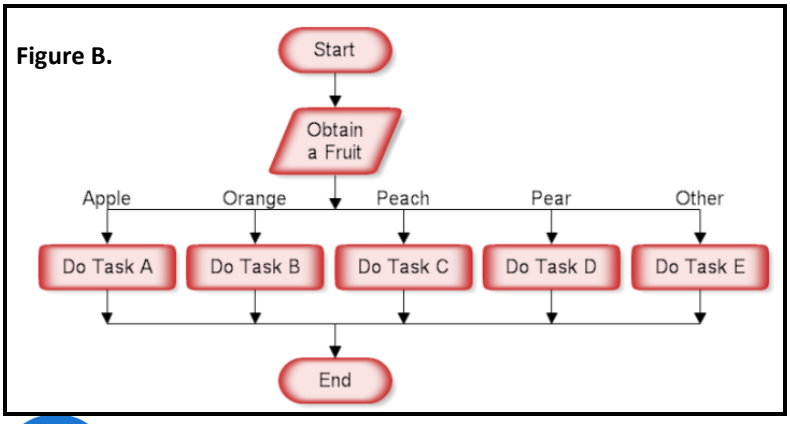


I. Multiple Choice. Select the letter of the correct answer.

1. What type of loop was used in Figure A?
   A. Do While loop b. While loop c. For loop d. Case
2. What is the value of `i` during the first loop in figure A?
   a. 8 b. 5 c. 2 d. none of the above
3. What is the last value of `i` that returns the statement true in Figure A?
   a. 9 b. 5 c. 8 d. none of the above
4. What type of Selection structure was used in Figure B?
   a. Sequence b. Repetition c. Switch d. Loop
5. Which one of the following is the CORRECT variable in figure B?
   a. Orange b. Peach c. Fruit d. Apple
6. How many cases are there in figure B?
   A. 4 b. 5 c. 6 d. none of the above

For numbers 7-10, identify the parts of the loop structure labeled A, B, C, D in Figure A.

Figure B.



References & Photo Credits

https://towardsdatascience.com/essential-programming-control-structures
https://www.academia.edu/24372685/CHAPTER_4_CONTROL_STRUCTURES_I_SELECTION
https://www.scribd.com/document/437581359/Control-Structure-and-Method
https://www.slideshare.net/AniLK0221/control-structure-c
https://towardsdatascience.com/essential-programming-control-structures-2e5e73285df4
https://www.tpub.com/progintro/1a.htm
https://slideplayer.com/slide/8229951/
https://whyisdifference.com/technology/difference-between-while-and-do-while-loop.html
https://beginnersbook.com/2017/08/cpp-for-loop/
https://www.tenouk.com/clabworksheet/labworksheet7_1.html
https://www.geeksforgeeks.org/nested-switch-case/
https://otrasteel.blogspot.com/2018/01/35-cannot-jump-from-switch-statement-to.html
https://www.slideshare.net/chaihidalgo/switch-case-and-looping-14700449
https://www.oreilly.com/library/view/computer-science-programming/9781449356835/fivedot2_while_loops.html
http://www.trytoprogram.com/c-programming/c-programming-for-loop/
https://www.rff.com/structured_flowchart.php
https://www.slideshare.net/jinmike008/switch-statements
https://press.rebus.community/programmingfundamentals/chapter/case-control-structure/
FLOWCHART SYMBOLS. https://www.pinterest.ph/pin/470415123569086615/visual-search/?x=14&y=13&w=453&h=426

Prepared by: MARICAR R. PORNEL, Teacher 1

GLOSSARY:

FLOWCHART . It is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem.

Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Basic symbols of Flowchart

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Note. These are the flowchart symbols used in our examples.

Building blocks . Building blocks may refer to the components of a particular programming language. In Java, one example component of a building blocks are statements. Statements are action or sequence of actions, given as a command in code. A statement ends with a semi-colon (;). An example structure of Java is shown below.

```
public static void main(String[] args)

  {

    // statements

  }
```