Algorithms

consist of a set of instructions that are carried out (performed) one after another. Sometimes there may be more than one path (or set of steps) that can be followed. At this point, a decision needs to be made. This point is known as selection. Depending on the answer given, the algorithm will follow certain steps and ignore others.

Why is selection important?

Selection allows us to include more than one path through an algorithm.

selection is usually represented by the instructions **IF**, **THEN** and **ELSE**.

- IF represents the question
- THEN points to what to do if the answer to the question is true
- ELSE points to what to do if the answer to the question is false

create a conditional control:

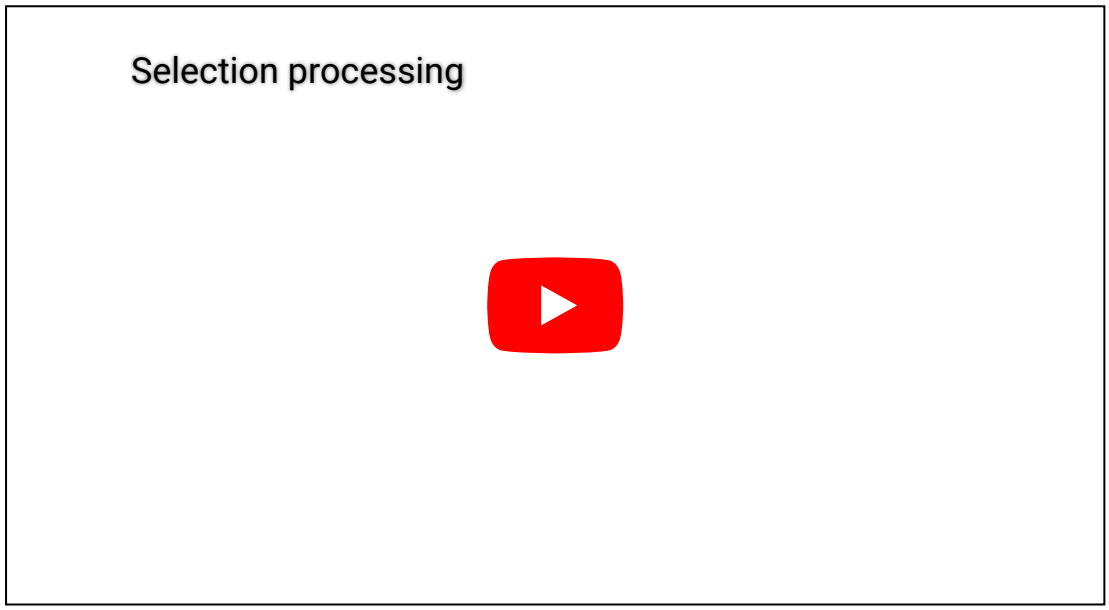
When you have to create a conditional control in your program with more than 3 or 4 conditions, it becomes complicated with if- else, here, a SWITCH statement became a quite useful.

The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.

However, the switch statement can replace an If-Else statement in the case where we compare a

variable to several integral values. You can use the switch statement only with the equals operation.

```
SWITCH (n) DO
    CASE 1 : // code to be executed if n = 1;
             BREAK; // BREAK is used to skip the other cases
    CASE 2 : // code to be executed if n = 2;
    DEFAULT :  // code to be executed if n doesn't match any cases
END_SWITCH
```



Selection processing in practice

In this slide, we are going to see a real example using the conditional statement, the IF-ELSE, and the Switch.

Let's start with the IF statement.

Our problem to solve is that the ticket price is variable according to the passenger age. if the age is under 16 then the ticket price will be half of the real price.

If the age over 16 then the ticket price will be the real price.

The statement will be like below:

```
/* *********** Bus tickets ******* */

/*
ask how old are you
```

```
IF you are under 16, THEN pay half ticket
ELSE pay full ticket
*/
// first solution
ticket_price := 20 ;
IF (age <=16) THEN
   ticket_price := 10 ;
END_IF

// Second solution
IF (age <= 16) THEN
   ticket_price := 10 ;
ELSE
   ticket_price := 20 ;
END_IF
```

Well let's try to make it a little harder, we'll have three test cases:

a case where the age is under 10, the passenger will pay only 20% of the real price;

a case where the age is between 10 and 16, the passenger will pay 50% of the ticket price.

a case where the age is above 16, the passenger will pay a full ticket.

The solution will look like:

```
IF (age<=10) THEN
   ticket_price := 4 ;
ELSE_IF (age<=16) THEN
/*  in this case, we don't specify whether the age is not under 10
because in this case it will satisfy the first condition
and execute the first block of instruction */
   ticket_price := 10 ;
ELSE
   ticket_price := 20 ;
END_IF
```

Selection processing in practice

Now let's try to resolve another problem using the switch statement.

Our problem is, we are going to print the opening hours of a zoo. The problem here is that the zoo opens at a different time according to the season. Example if it is summer the door get opened at 10:00 and close at 20:00

if it is winter it open at 10:00 and closed at 16:00

Now since we only have four seasons, we are going to represent these season as shown below

1 refer to winter

2 refer to spring

3 refer to summer

4 refer to autumn

And here is the solution to this problem:

```
/* **** Zoo time **** */
/*
1 refer to winter
2 refer to spring
3 refer to summer
4 refer to autumn
*/
SWITCH (season) DO
   case 1 : Write("10h00 to 16h00"); BREAK;
   case 2 : Write("10h00 to 18h00"); BREAK;
   case 3 : Write("10h00 to 20h00"); BREAK;
   case 4 : Write("10h00 to 16h00"); BREAK;
   default : Write("Wrong number") // optional use of BREAK;
END_SWITCH
```

Now the thing that we should notice in this code is the existence of the BREAK keyword. Let's suppose that the season is 3, and we remove the BREAK keyword when we execute the program display "10h00 to 20h00" "10h00 to 16h00" "Wrong number"

That's due to how the switch works if in any case, the condition is true then it will execute all the rest of the cases. and to make it right we add the BREAK keyword.