បណ្ឌិត្យសភាបច្ចេកវិទ្យាឌីជីថលកម្ពុជា
Cambodia Academy of Digital Technology

វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

Week 06

Control Statements – Selection

Prepared by: **Leangsiv HAN**

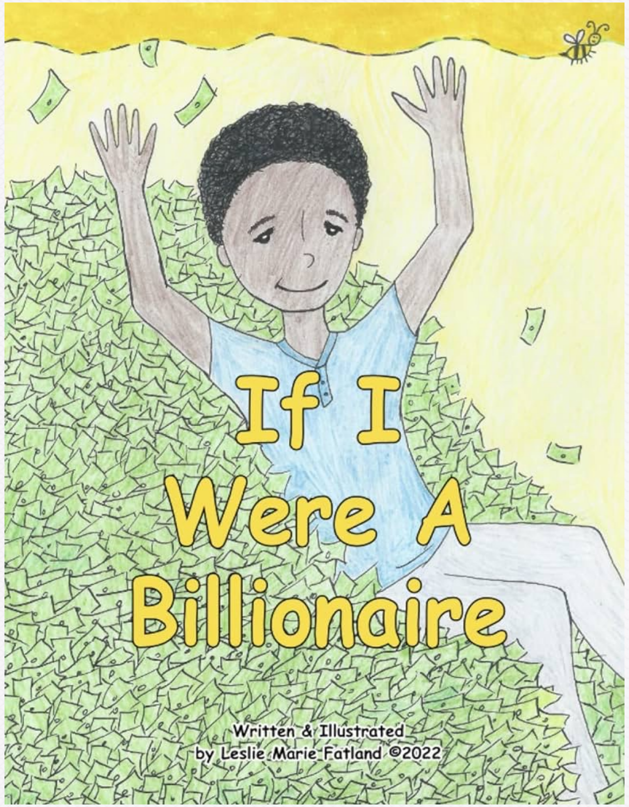
20th March 2024

Last Week

You have learned about:

- ❑ The definition of flowchart.
- ❑ flowchart symbols.
- ❑ The technique for drawing flowcharts.
- ❑ The rules for creating flowcharts.

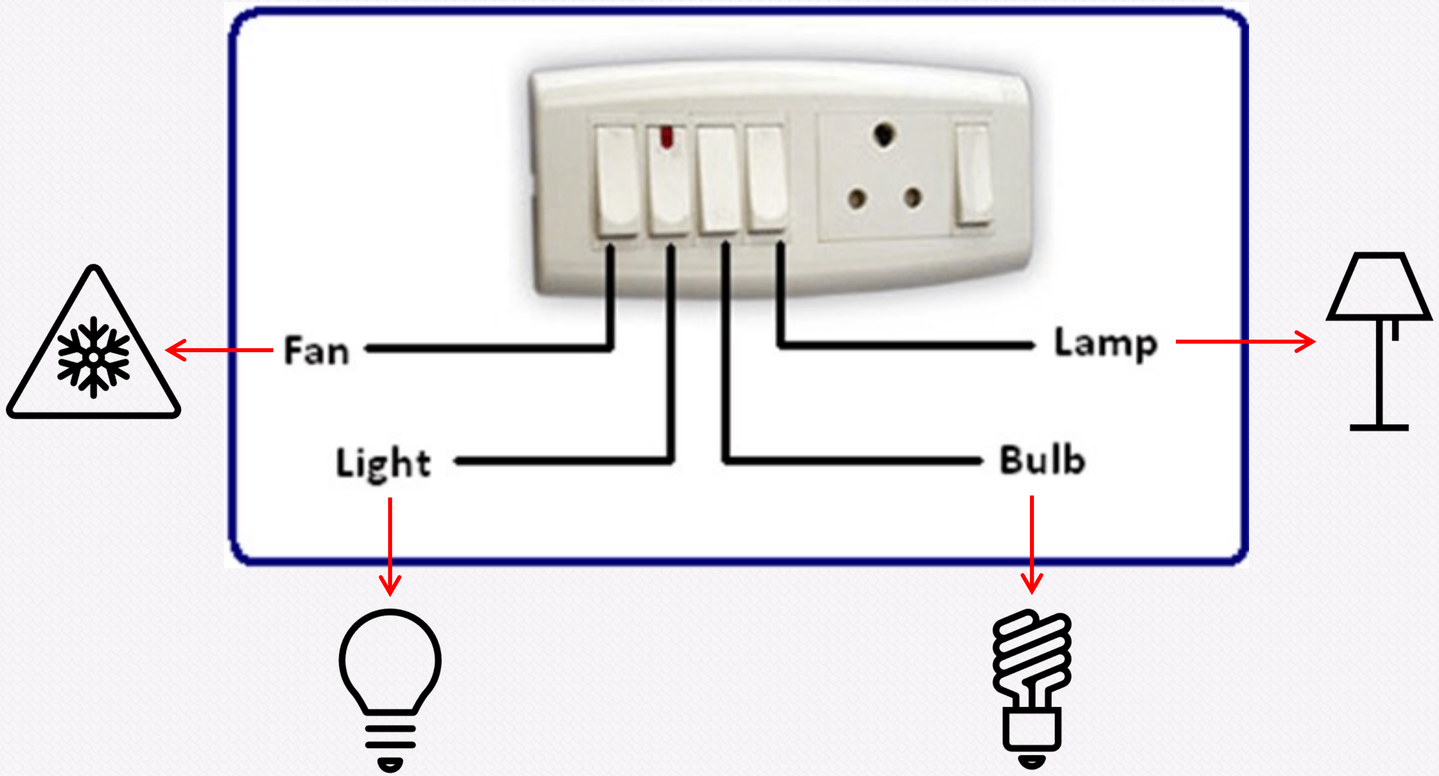
Let’s get started with this image!



How about this image?

If I get atleast 10 rupees,
I will buy chocolate
else I will buy candy

Last Image!



Learning Objectives

By the end of this lesson, you will be to:

- ❑ Define control statements and their programs.
- ❑ Implement selection statements, including if, if-else, ternary operator, else-if, and switch statements.

Control Statement

- In simple words, **Control statements** in C help the computer execute a certain *logical statement* and decide whether to *enable the control of the flow* through a certain set of statements or not.
- Also, it is used to direct the execution of statements under certain *conditions*.

Program Control Statements

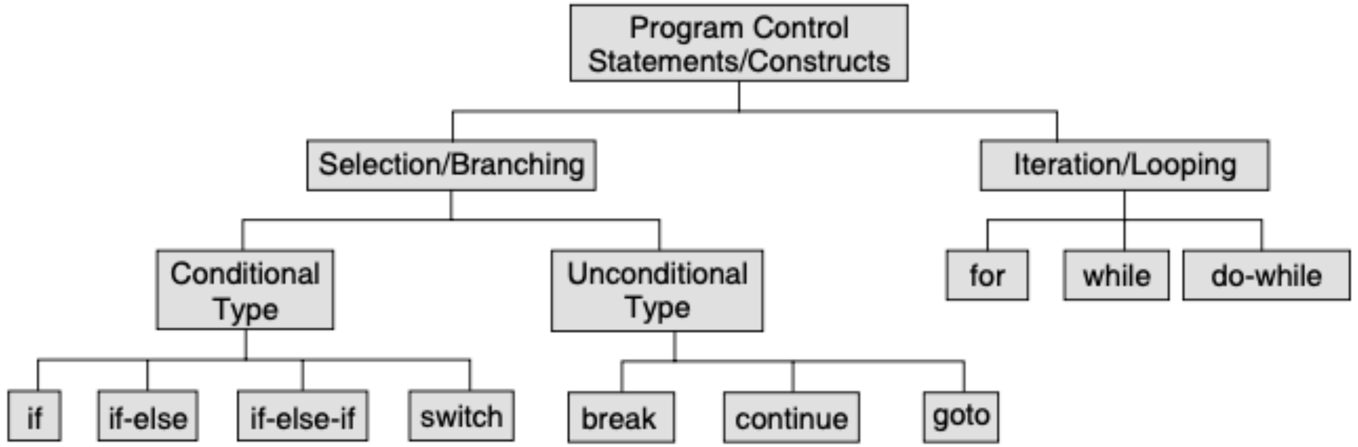


Fig. 10.1 Program control statements/constructs in C

Selection Statement

- A **selection statement** is a control statement that allows choosing between *two or more execution* paths in a program.
- The selection statements in C are the **if statement**, the **if-else statement**, and **the switch statement**.
- These statements allow us to decide which statement to execute next.

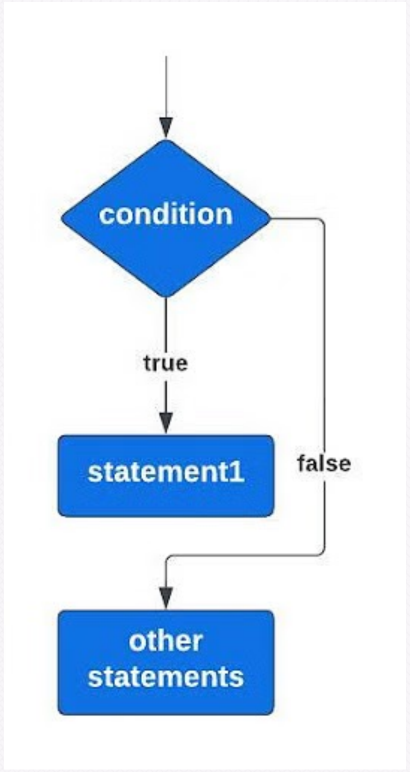
Simple if Statement (1/2)

- Simple **if statements** are carried out to perform some operation when the condition is only true. If the condition of the if statement is true then the statements under the if block is executed else the control is transferred to the statements outside the if block.
- **Syntax:**

```
if (condition) {
    // block of code to be executed if the condition is true
}
```

Simple if Statement (2/2)

- **Flowchart:**



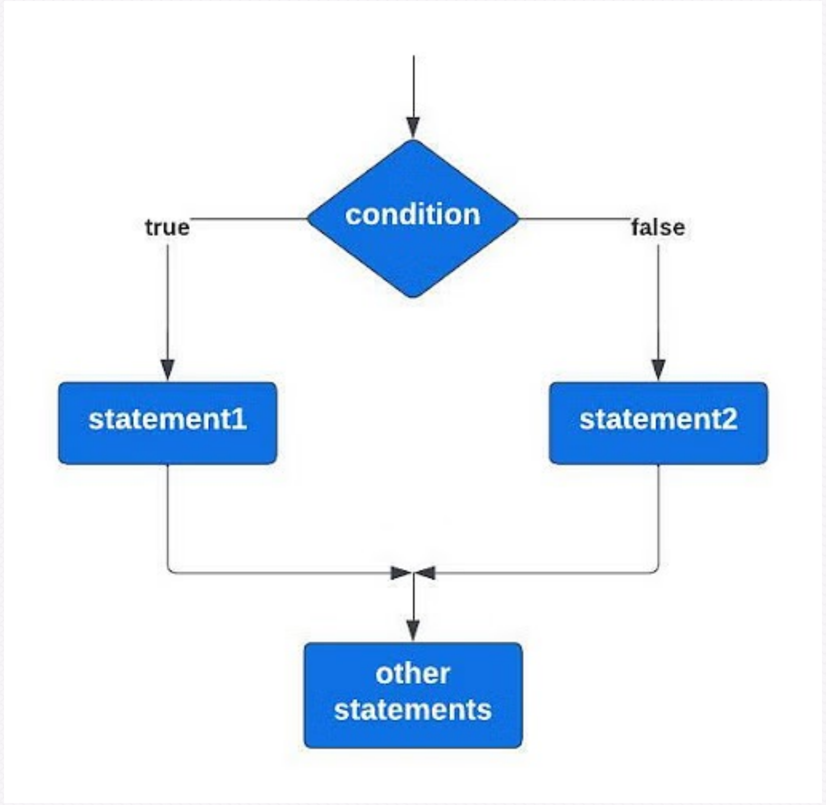
if-else Statement (1/2)

- In some situations, you may have to execute statements based *on true or false* under certain conditions, therefore; you use **if-else statements**. If the condition is true, then if block will be executed otherwise the else block is executed.
- **Syntax:**

```
if (condition) {
    // block of code to be executed if the condition is true
} else {
    // block of code to be executed if the condition is false
}
```

if-else Statement (2/2)

- **Flowchart:**



Ternary Operator (1/2)

- There is also a short-hand if else, which is known as the **ternary operator** because it consists of three operands.
- It can be used to *replace multiple lines* of code with a single line.
- *It is often used to replace simple if-else statements.*
- **Syntax**:

```
variable = (condition) ? expressionTrue : expressionFalse;
```

Ternary Operator (2/2)

Instead of Writing:

```
int time = 20;
if (time < 18) {
  printf("Good day.");
} else {
  printf("Good evening.");
}
```

You can simply write:

```
int time = 20;
(time < 18) ? printf("Good day.") : printf("Good evening.");
```

else-if Statement (1/3)

- The **else-if statements** contain *multiple else-if*, when either of the condition(s) is true the statements under that particular "if" will be executed otherwise the statements under the else block will be executed.
- Suppose the "if" condition is true, statements under "if" will be executed else the other "if" condition is tested, and if that condition is true statements under that particular "if" will be executed. This process will repeat as long as the else-if's are present in the program.

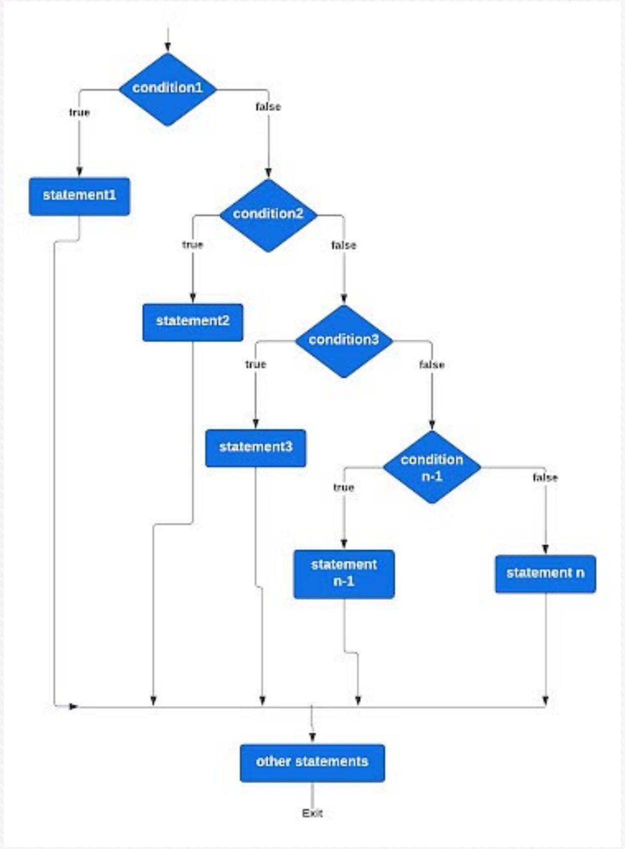
else-if Statement (2/3)

- **Syntax:**

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and condition2 is true
} else {
  // block of code to be executed if the condition1 is false and condition2 is
false
}
```

else-if Statement (3/3)

- **Flowchart:**



switch Statement (1/3)

- As per the value of the switch expression, the **switch statement** will allow *multi-way branching*.
- Depending on the expression, the control is transferred to that particular **case label** and executed the statements under it. If none of the cases are matched with the *switch expression*, then the *default statement is executed*.

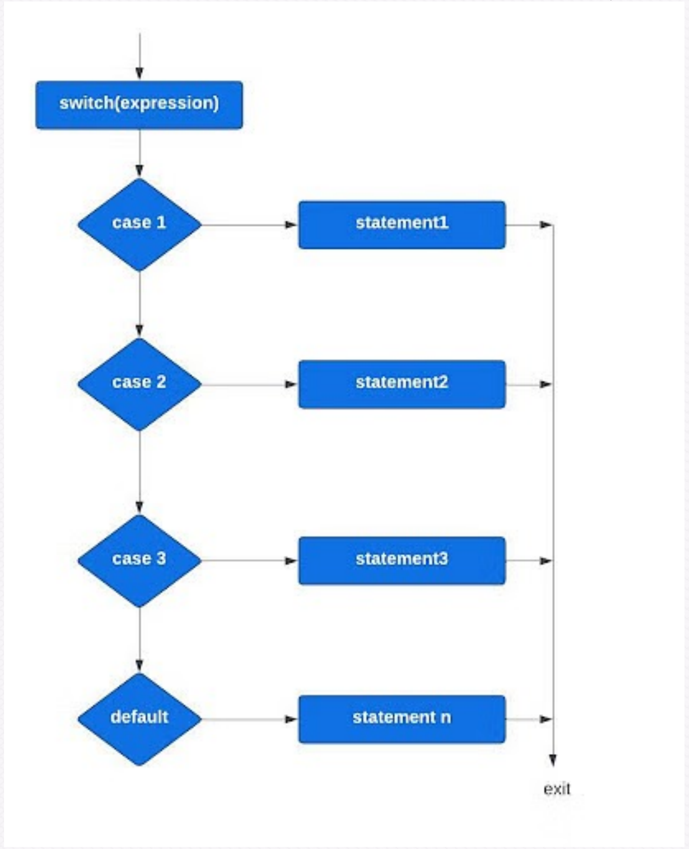
switch Statement (2/3)

- **Syntax:**

```
switch (expression) {
 case x:
   // code block
   break;
 case y:
   // code block
   break;
 default:
   // code block
}
```

switch Statement (3/3)

- **Flowchart:**



Self-study

- What is Nested if and how to use it?
- What is the difference between if-else if and switch?
- Can you provide some real-life examples from the lesson today?

Key Takeaways

You are now able to:

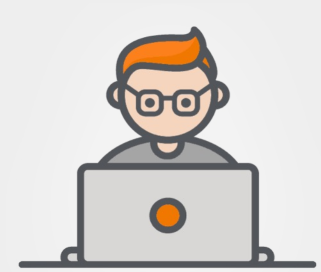
- ✓ Define control statements and their programs.
- ✓ Implement selection statements, including if, if-else, ternary operator, else-if, and switch statements.

References

- Dey, P., & Ghosh, M. (2013). *Computer fundamentals and programming in C*.
- S, H. S. (2022, May 13). *Control statements in C: An ultimate guide | Simplilearn*. Simplilearn.com. https://www.simplilearn.com/tutorials/c-tutorial/conditional-control-statements-in-c
- *C if ... Else conditions*. (n.d.). W3Schools Online Web Tutorials. https://www.w3schools.com/c/c_conditions.php

Thank you !

Questions or Feedbacks?



Contact Me via:

✉ leangsiv.han@cadt.edu.kh

t @leangsiv

in leangsivhan