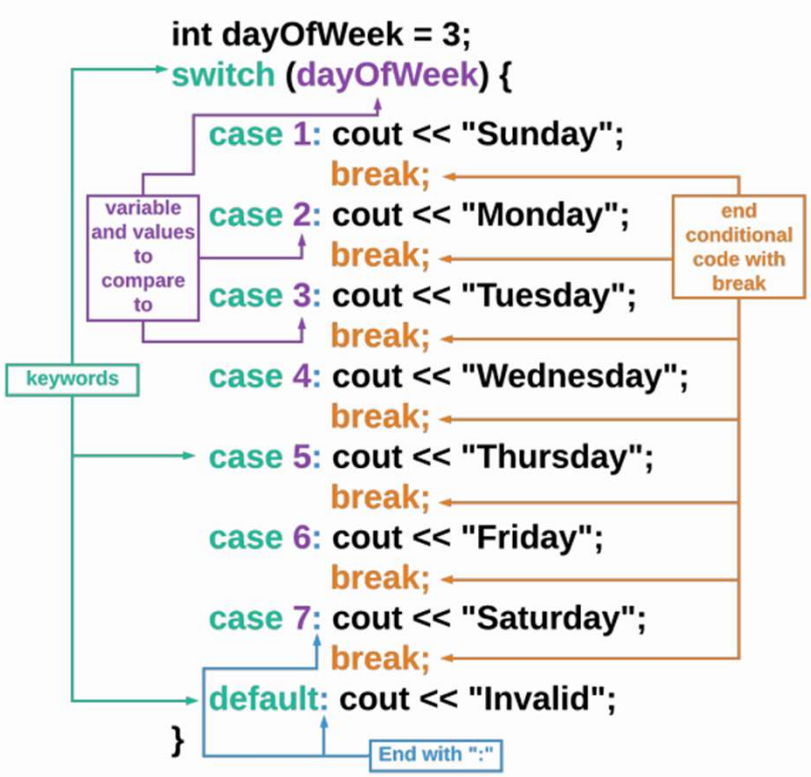
Switch case Statement

The switch case statement is a way to make a decision with multiple possible outcomes.

It evaluates an expression and matches it with a particular value, then it executes set of statements associated with that value

Syntax of Switch-case



Rules for Writing a switch case statement

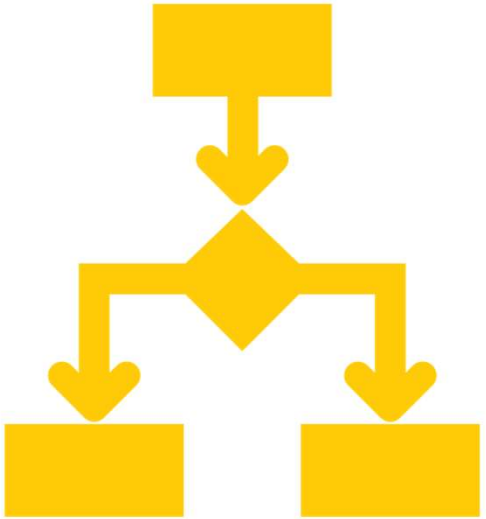
- Start with `switch` followed by the variable that is going to be tested in parentheses `()`.
- All of the `cases` are surrounded by a set of curly braces `{}`.
- Each `case` is followed by a *numerical* value and a colon `:`.
- After each `:`, write the code that should run if the variable is equal to that case's value.
- After each section of code per case, include `break;`.
- As the very last case, use `default:` to specify what should happen if none of the above cases are true.

Challenge

- Input a month in number from 1-12 and display the relevant day is string form.

Switch case vs If-Else-if

- Switch case and else-if both allows to check for multiple outcomes
- Then why switch case



#1: Else If is used for ranges of values - Switch Case is for specific values

`switch case` can only check for equality (e.g. `num == 5`), so if you need to check for a range of values (e.g. `num > 50 && num <= 60`), use `else If` instead.

```
int grade = 62;
int letterGrade = grade / 10;
switch (letterGrade) {
  case 10: case 9: cout << "A";
           break;
  case 8: cout << "B";
           break;
  case 7: cout << "C";
           break;
  case 6: cout << "D";
           break;
  default: cout << "F";
}
```

```
int grade = 62;
if (grade < 60) {
  cout << "F"; }
else if (grade < 70) {
  cout << "D"; }
else if (grade < 80) {
  cout << "C"; }
else if (grade < 90) {
  cout << "B"; }
else if (grade <= 100) {
  cout << "A"; }
```

#2: Else If is used for handling multiple variables

Switch case can only compare against values - not variables.

Compare the inputted day of the week with the current day of the week, you would need to use else if.

- Switch case can handle values (dayOfWeek == "Sunday") but not variables (dayOfWeek == today).

Rules of the switch case statement in C++

- The case value must be either int or char type.
- There can be any number of cases.
- No duplicate case values are allowed.
- Each statement of the case can have a break statement. It is optional.
- The default Statement is also optional.

Advantages of Switch-case

- Faster execution speed.
- Easier to read than if else if.

Disadvantages of Switch-case

1. Switch case can only evaluate int or char type.
2. No support for logical expressions.
3. Have to keep in mind to add a break in every case.

Important points to consider when using switch-case

- No duplicate Case Values
- Break statement is optional

Nested Switch-Case

```cpp
#include <iostream>
using namespace std;

int main()
{

    int var1 = 1;
    int var2 = 0;

    // outer switch
    switch (var1) {
      case 0:
          cout << "Outer Switch Case 0\n";
          break;
      case 1:
          cout << "Outer Switch Case 1\n";
          // inner switch
          switch (var2) {
          case 0:
              cout << "Inner Switch Case 0\n";
              break;
          }
          break;
      default:
          cout << "Default Case of Outer Loop";
          break;
    }

    return 0;
}
```