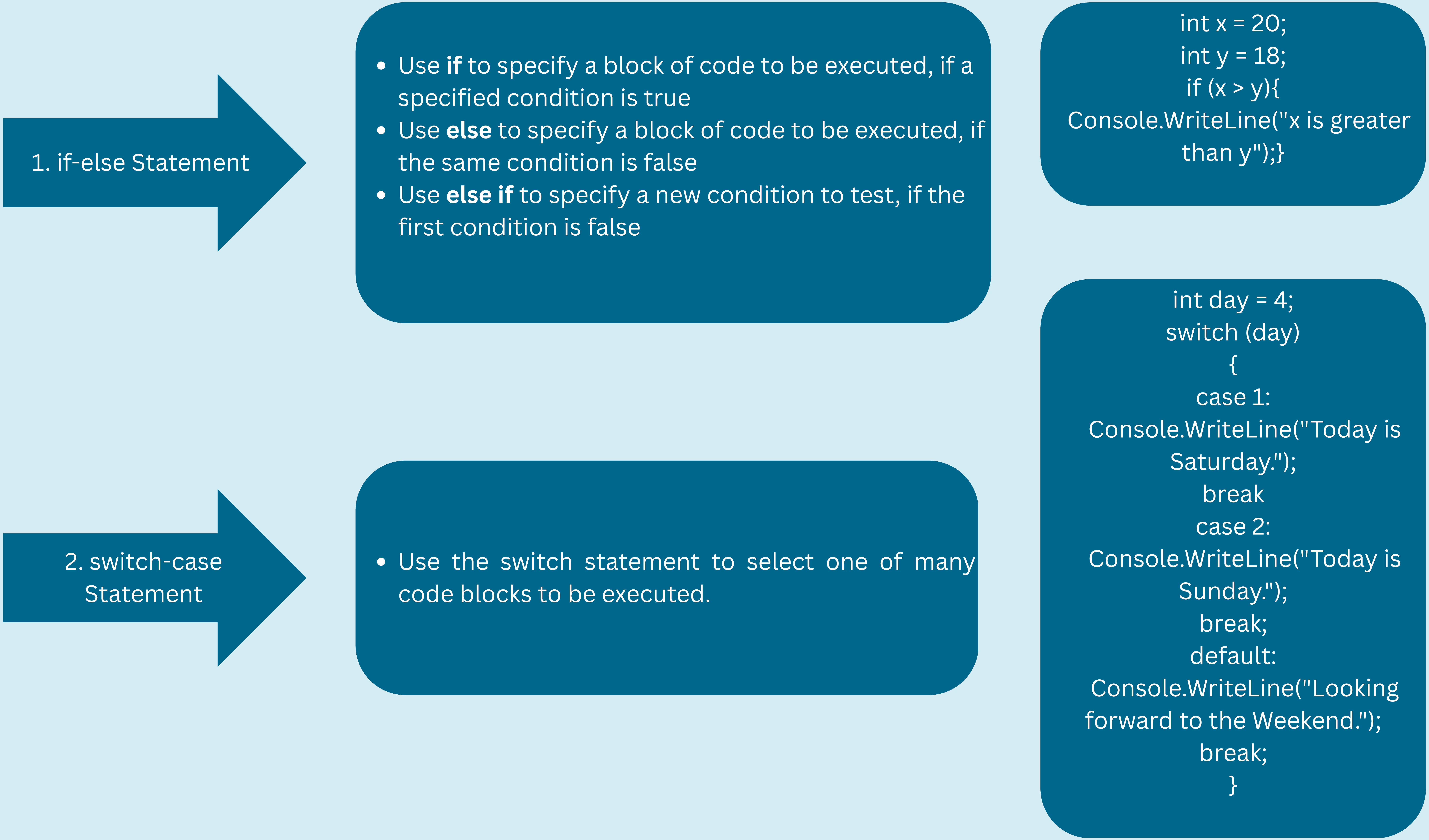
Conditional Statements



1. if-else Statement

- Use **if** to specify a block of code to be executed, if a specified condition is true
- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false

```
int x = 20;
int y = 18;
if (x > y){
Console.WriteLine("x is greater
than y");}
```

2. switch-case Statement

- Use the switch statement to select one of many code blocks to be executed.

```
int day = 4;
switch (day)
{
case 1:
Console.WriteLine("Today is
Saturday.");
break
case 2:
Console.WriteLine("Today is
Sunday.");
break;
default:
Console.WriteLine("Looking
forward to the Weekend.");
break;
}
```

Loops

Loops can execute a block of code as long as a specified condition is reached.
Loops are handy because they save time, reduce errors, and they make code more readable.

1. For Loop

When you know exactly how many times you want to loop through a block of code, use the for loop

ex:

```
for (int i = 0; i < 8; i++){
 Console.WriteLine(i);}
```

2. While Loop

The while loop loops through a block of code as long as a specified condition is True.

ex:

```
int i = 0;
while (i < 8)
{
Console.WriteLine(i);
i++;
}
```

3. Do-While Loop

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

ex:

```
int i = 0;
do
{
 Console.WriteLine("i = {0}", i);
i++;
}
while (i < 5);
```