

## 3.2 If-else

### If statements

An **if** statement executes a group of statements if an expression is true. Braces surround the if branch's statements. **Braces** {}, sometimes redundantly called curly braces, represent a grouping, such as a grouping of statements. Note: {} are braces, [] are brackets.

Good practice is to indent a branch's statements, using a consistent number of spaces. This material indents 3 spaces.

#### PARTICIPATION ACTIVITY

3.2.1: if statement: Hotel discount.



Start



2x speed

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

int main() {
    int hotelRate;
    int userAge;

    hotelRate = 155;

    cout << "Enter age: ";
    cin >> userAge;

    if (userAge > 65) {
        hotelRate = hotelRate - 20;
    }

    cout << "Your hotel rate: ";
    cout << hotelRate << endl;

    return 0;
}
```

Enter age: 68  
Your hotel rate: 135

```
hotelRate = 155;
cout << "Enter age: ";
cin >> userAge;
if (userAge > 65) {
    hotelRate = hotelRate - 20;
}
cout << "Your Hotel rate: ";
cout << hotelRate << endl;
```

Memory	
95	
96	135 hotelRate
97	68 userAge
98	

[Feedback?](#)

## ACTIVITY

## 3.2.2: If statement.

What is the final value of numItems?

1) `bonusVal = 19;`  
`numItems = 1;`  
`if (bonusVal > 10) {`  
    `numItems = numItems + 3;`  
`}`

[Check](#)[Show answer](#)

2) `bonusVal = 0;`  
`numItems = 1;`  
`if (bonusVal > 10) {`  
    `numItems = numItems + 3;`  
`}`

[Check](#)[Show answer](#)[Feedback?](#)

## If-else statement

An **if-else** statement executes one group of statements when an expression is true, and another group of statements when the expression is false.

Construct 3.2.1: If-else statement.

```
// Statements that execute before the branches

if (expression) {
    // Statements that execute when expression is true (first branch)
}
else {
    // Statements that execute when expression is false (second branch)
}

// Statements that execute after the branches
```

[Feedback?](#)

In the example below, if a user inputs an age less than 25, the statement `insurancePrice = 4800` executes. Otherwise, `insurancePrice = 2200` executes.

### PARTICIPATION ACTIVITY

#### 3.2.3: if-else statement: Car insurance.



Start



2x speed

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

int main() {
    int userAge;
    int insurancePrice;

    cout << "Enter age: ";
    cin >> userAge;

    if (userAge < 25) {
        insurancePrice = 4800;
    }
    else {
        insurancePrice = 2200;
    }

    cout << "Annual price: $";
    cout << insurancePrice << endl;

    return 0;
}
```

#### Memory

95		
96	2200	insurancePrice
97	40	userAge
98		

Enter age: 22  
Annual price: \$ 4800

Enter age: 40  
Annual price: \$ 2200

[Feedback?](#)

## Car insurance prices

(*Car insurance prices* for drivers under 25 are higher because 1 in 6 such drivers are involved in an accident each year, vs. 1 in 15 for older drivers. Source: [www.census.gov](http://www.census.gov), 2009).

### PARTICIPATION ACTIVITY

#### 3.2.4: If-else statements.



1) What is the final value of numItems?



```
bonusVal = 5;  
if (bonusVal < 12) {  
    numItems = 100;  
}  
else {  
    numItems = 200;  
}
```

**Check**[Show answer](#)

2) What is the final value of numItems?



```
bonusVal = 12;  
if (bonusVal < 12) {  
    numItems = 100;  
}  
else {  
    numItems = 200;  
}
```

**Check**[Show answer](#)

3) What is the final value of numItems?



```
bonusVal = 15;  
numItems = 44;  
if (bonusVal < 12) {  
    numItems = numItems + 3;  
}  
else {  
    numItems = numItems + 6;  
}  
numItems = numItems + 1;
```

**Check**[Show answer](#)

4) What is the final value of bonusVal?



```
bonusVal = 11;  
if (bonusVal < 12) {  
    bonusVal = bonusVal + 2;  
}  
else {  
    bonusVal = bonusVal + 10;  
}
```

**Check**[Show answer](#)

5) What is the final value of bonusVal?



```
bonusVal = 11;
if (bonusVal < 12) {
    bonusVal = bonusVal + 2;
    bonusVal = 3 * bonusVal;
}
else {
    bonusVal = bonusVal + 10;
}
```

**Check**[Show answer](#)[Feedback?](#)**PARTICIPATION  
ACTIVITY**

## 3.2.5: Writing an if-else statement.

Translate each description to an if-else statement as directly as possible. Use {}. (Not checked, but please indent a branch's statements some consistent number of spaces, such as 3 spaces).

- 1) If userAge is greater than 62, assign itemDiscount with 15. Else, assign itemDiscount with 0.

**Check**[Show answer](#)

- 2) If numPeople is greater than 10, execute groupSize = 2 \* groupSize. Otherwise, execute groupSize = 3 \* groupSize and numPeople = numPeople - 1.

**Check****Show answer**

- 3) If numPlayers is greater than 11, execute teamSize = 11. Otherwise, execute teamSize = numPlayers. Then, no matter the value of numPlayers, execute teamSize = 2 \* teamSize.

**Check****Show answer****Feedback?****CHALLENGE  
ACTIVITY**

3.2.1: Enter the output for the if-else branches.

**Start**

Type the program's output.

ck

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

int main() {
    int numApples;

    numApples = 5;

    if (numApples < 7) {
        cout << "c" << endl;
    }
    else {
        cout << "e" << endl;
    }

    cout << "k" << endl;

    return 0;
}
```

1

2

Check

Next

[Feedback?](#)**CHALLENGE  
ACTIVITY**

## 3.2.2: Basic if-else expression.



Start



1

Write an expression that will cause the following code to print "greater than 20" if the value of userAge is greater than 20.



2



3

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int userAge;
6
7     cin >> userAge; // Program will be tested with values: 19, 20, 21, 22.
8
9     if (/* Your code goes here */) {
10         cout << "greater than 20" << endl;
11     }
12     else {
13         cout << "20 or less" << endl;
14     }
15
16     return 0;
17 }
```

1

2

3

Check

Next

[Feedback?](#)CHALLENGE  
ACTIVITY

## 3.2.3: Basic if-else.

[Jump to level 1](#)

1



2

Write an if-else statement for the following:

If numDifference is greater than -16, execute totalDifference = -10. Else, execute totalDifference = numDifference.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int totalDifference;
6     int numDifference;
7
8     cin >> numDifference; // Program will be tested with values: -13 -14 -15 -16.
9
10    /* Your code goes here */
11    if (numDifference > -16){
12        totalDifference = -10;
13    }
14    }else{
15        totalDifference = numDifference;
16    }
17
18    cout << totalDifference << endl;
19
20    return 0;
21 }
```

1

2

Check

Next

**Done.** Click any level to practice more. Completion is preserved.

✓ If numDifference is greater than -16, the expression evaluates to true and totalDifference is assigned with -10. Else, the expression evaluates to false and totalDifference is assigned with numDifference.

✓ 1: Compare output ^

Input -13

Your output -10



## ✓ 2: Compare output ^

Input -14

Your output -10

## ✓ 3: Compare output ^

Input -15

Your output -10

## ✓ 4: Compare output ^

Input -16

Your output -16

[Feedback?](#)

## Multi-branch if-else statements

An If-else statement can be extended to have three (or more) branches. Each branch's expression is checked in sequence. As soon as one branch's expression is found to be true, that branch's statement execute (and no subsequent branch is considered). If no expression is true, the else branch executes.

Construct 3.2.2: Multi-branch if-else statement. Only 1 branch will execute.

```
if (expression1) {  
    // Statements that execute when expression1 is true  
    // (first branch)  
}  
else if (expression2) {  
    // Statements that execute when expression1 is false and expression2 is true  
    // (second branch)  
}  
else {  
    // Statements that execute when expression1 is false and expression2 is false  
    // (third branch)  
}
```

[Feedback?](#)

The **equality operator** `==` evaluates to true if the left side and right side are equal. Ex: If `numYears` holds the value 10, then the expression `numYears == 10` evaluates to true.

Note that the equality operator is `==`, not `=`.

Figure 3.2.1: Multi-branch if-else example: Anniversaries.

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

int main() {
    int numYears;

    cout << "Enter number years married: ";
    cin >> numYears;

    if (numYears == 1) {
        cout << "Your first year -- great!" << endl;
    }
    else if (numYears == 10) {
        cout << "A whole decade -- impressive." << endl;
    }
    else if (numYears == 25) {
        cout << "Your silver anniversary -- enjoy." << endl;
    }
    else if (numYears == 50) {
        cout << "Your golden anniversary -- amazing." << endl;
    }
    else {
        cout << "Nothing special." << endl;
    }

    return 0;
}
```

```
Enter number years married: 10
A whole decade -- impressive.

...

Enter number years married: 25
Your silver anniversary -- enjoy.

...

Enter number years married: 30
Nothing special.

...

Enter number years married: 1
Your first year -- great!
```

[Feedback?](#)

#### PARTICIPATION ACTIVITY

#### 3.2.6: Multi-branch if-else statements.



What is the final value of `employeeBonus` for each given value of `numSales`?

```

if (numSales == 0) {
    employeeBonus = 0;
}
else if (numSales == 1) {
    employeeBonus = 2;
}
else if (numSales == 2) {
    employeeBonus = 5;
}
else {
    employeeBonus = 10;
}

```

1) numSales is 2

Check

Show answer

2) numSales is 0

Check

Show answer

3) numSales is 7

Check

Show answer

[Feedback?](#)

## Common errors

When a branch has a single statement, the braces are optional, but good practice always uses the braces. Always using braces even when a branch only has one statement prevents the common error of mistakenly thinking a statement is part of a branch.

### PARTICIPATION ACTIVITY

3.2.7: Common error when omitting braces.

1 2 2x speed

```

if (numSales < 20)    15 < 20
    salesBonus = 0;
else
    totBonus = totBonus + 1;
    salesBonus = 20;

```

Memory		
95		
96	15	numSales
97	20	salesBonus
98	2	totBonus

Braces aren't used, so the else branch's only statement is `totBonus = totBonus + 1`. But, `salesBonus = 20;` should also be part of the else branch.

[Feedback?](#)**PARTICIPATION  
ACTIVITY**

## 3.2.8: Braces are important.

Omitting braces is a common source of errors. What is the final value of `numItems`?

1) 

```
numItems = 0;
bonusVal = 19;
if (bonusVal > 10)
    numItems = bonusVal;
numItems = numItems + 1;
```

**Check**[Show answer](#)

2) 

```
numItems = 0;
bonusVal = 5;
if (bonusVal > 10)
    // Need to update bonusVal
    numItems = bonusVal;
numItems = numItems + 1;
```

**Check**[Show answer](#)

3) 

```
numItems = 0;
bonusVal = 5;
if (bonusVal > 10)
    // Update bonusVal
    bonusVal = bonusVal - 1;
    numItems = bonusVal;
numItems = numItems + 1;
```

[Check](#)[Show answer](#)[Feedback?](#)**CHALLENGE  
ACTIVITY**

## 3.2.4: If-else statement: Fix errors.



Re-type the code and fix any errors. The code should convert non-positive numbers to 1.

```
if (userNum > 0)
    cout << "Positive." << endl;
else
    cout << "Not positive, converting to 1." << endl;
    userNum = 1;

cout << "Final: " << userNum << endl;
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int userNum;
6
7     cin >> userNum;
8
9     /* Your solution goes here */
10    if (userNum > 0)
11        cout << "Positive." << endl;
12    else{
13        cout << "Not positive, converting to 1." << endl;
14        userNum = 1;
15    }
16
17    cout << "Final: " << userNum << endl;
18
19    return 0;
20 }
```

[Run](#)

✓ All tests passed

✓ Testing with userNum = -5.

Your output

```
Not positive, converting to 1.
Final: 1
```

✓ Testing with userNum = 99.

Your output

```
Positive.
Final: 99
```

✓ Testing with userNum = -1.

Your output

```
Not positive, converting to 1.  
Final: 1
```

[Feedback?](#)