



FACULTY OF ENGINEERING AND TECHNOLOGY

COMPUTER SCIENCE DEPARTMENT

COMP1310 Introduction to Computer and Computing Ethics

STUDENTS-HUB.com

SELECTION STRUCTURES: IF AND SWITCH

Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022 Uploaded By: anonymous

STUDENTS-HUB.com

Control Structures

- A control structure is a combination of individual instructions into a single logical unite with one entry point and one exit point.
- They are usually bracketed by { and }
- A selection control structure is a control structure that chooses among alternative program statements.
- This means that there is a condition the guides the selection control structure on which alternative to choose.
- A condition is an expression that is either false or true.
- Note that false can be represented as 0, and true can be represented as a non-zero number, usually 1.

Relational and Equality Operators

- Conditions usually have one of the following forms:
 - variable relational-operator
 - variable equality-operator

variable/constant

variable/constant

Operator	Meaning	Туре
<	Less than	Relational
>	Greater than	Relational
<=	Less than or equal	Relational
>=	Greater than or equal	Relational
==	Equal to	Equality
!=	Not equal to	Equality

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Logical Operators

A logical expression is an expression that uses one or more logical operators.

Operator	Meaning
&&	AND
11	OR
!	NOT

Truth Tables - AND

operand1	operand2	operand1 && operand2
True	True	True
True	False	False
False	True	False
False	False	False

Truth Tables - OR

operand1	operand2	operand1 operand2
True	True	True
True	False	True
False	True	True
False	False	False

Truth Tables - NOT

operand	!operand
True	False
False	True

Operator Precedence - Updated

From the highest to the lowest

- Function calls
- Unary operators (! + -)
- * / %
- **•** + -
- < <= >= >
- == !=
- **■** &&
- **—** =

ST

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z		

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0		

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive		

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	T && T is T

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	T && T is T
x is outside the range z to y		

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	T && T is T
x is outside the range z to y	!(z <= x && x <= y)	

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	T && T is T
x is outside the range z to y	!(z <= x && x <= y)	!(T && T) is !T is F

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	T && T is T
x is outside the range z to y	!(z <= x && x <= y)	!(T && T) is !T is F
	z > x x > y	

Let's assume here that x = 3.0, y = 4.0, and z = 2.0

English Condition	Logical Expression	Evaluation
x and y are greater than z	x > z && y > z	T && T is T
x is equal to 1.0 or 3.0	x == 1.0 x == 3.0	F T is T
x is in the range z to y inclusive	z <= x && x <= y	T && T is T
x is outside the range z to y	!(z <= x && x <= y)	!(T && T) is !T is F
	z > x x > y	F F is F

Uploaded By: anonymous

S

Logical Assignment

We can assign the result of a conditional statement to a variable, such as an integer.

- The value that will be stored in the variable will be:
 - 1 if the conditional statement is true
 - 0 if the conditional statement is false
- For example, if we want to print the value of a conditional statements that checks if a grade is a passing grade or not.

Logical Assignment – cont.

```
1 #include <stdio.h>
2
3 void main() {
4     printf("Please enter a grade: ");
5     int grade;
6     scanf("%d", &grade);
7     int pass = (grade >= 60);
8     printf("Grade %d is a pass = %d.", grade, pass);
9 }
```

The *if* statement

- With if statements we can tell the compiler to do something specific only when some condition is true.
- if statements can be used to run one statement, or compound statements.
- if statements can have one alternative:
 - do something if the condition is true.
- or they can have two alternatives:
 - do something if the condition is true.
 - do something else if the condition is false.

if (condition)

something;

One alternative, single statement

Two alternatives, single statements

if (condition)

something;

else

something;

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022



Two alternatives, compound statements

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022



Two alternatives, compound statements

if statements can be made of any combination of alternatives and statements.

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

- Let's modify the example we used previously:
 - Write a program that reads a student's grade and prints out if the student passed or failed.
- The condition:

- Let's modify the example we used previously:
 - Write a program that reads a student's grade and prints out if the student passed or failed.
- The condition: the student's grade is larger than or equal to 60.

- Let's modify the example we used previously:
 - Write a program that reads a student's grade and prints out if the student passed or failed.
- The condition: the student's grade is larger than or equal to 60.
- If the condition is true:

- Let's modify the example we used previously:
 - Write a program that reads a student's grade and prints out if the student passed or failed.
- The condition: the student's grade is larger than or equal to 60.
- If the condition is true: print that the student passed.

- Let's modify the example we used previously:
 - Write a program that reads a student's grade and prints out if the student passed or failed.
- The condition: the student's grade is larger than or equal to 60.
- If the condition is true: print that the student passed.
- If the condition is false:
Example – print 'pass' or 'fail'

- Let's modify the example we used previously:
 - Write a program that reads a student's grade and prints out if the student passed or failed.
- The condition: the student's grade is larger than or equal to 60.
- If the condition is true: print that the student passed.
- If the condition is false: print that the student failed.

Example – print 'pass' or 'fail'

```
#include <stdio.h>
 2
 3
    void main() {
        printf("Please enter a grade: ");
 4
 5
        int grade;
 6
        scanf("%d", &grade);
 7
         if (grade >= 60)
8
             printf("The student passed.");
9
         else
10
             printf("The student failed.");
11
12
```

Example – multiple alternative conditions

Write a program that reads a student's average and prints out their letter grade based on the following criteria:

Average	Letter grade
$0 \le average < 60$	F
$60 \le average < 70$	D
$70 \leq average < 80$	C
$80 \le average < 90$	В
90 ≤ average	A

Example – multiple alternative conditions

```
#include <stdio.h>
     1
     2
     3
         void main() {
     4
              printf("Please enter a grade: ");
     5
              int grade;
     6
              scanf("%d", &grade);
     7
              if (grade < 60)
     8
                   printf("F");
     9
              else
    10
                   if (grade < 70)
                       printf("D");
    11
    12
                   else
    13
                        if (grade < 80)
    14
                            printf("C");
    15
                       else
    16
                            if (grade < 90)
    17
                                 printf("B");
    18
                            else
    19
                                 printf("A");
    20
Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022
```

Uploaded By: anonymous

S

DENTS-HUB.com

■ What is the output of the following program:

```
1 #include <stdio.h>
2
3 void main() {
4     int x = 0;
5     if (x)
6         printf("condition is true\n");
7         printf("this statement\n");
8     }
```

■ What is the output of the following program:

```
1 #include <stdio.h>
2
3 void main() {
4     int x = 0;
5     if (x)
6         printf("condition is true\n");
7         printf("this statement\n");
8     }
```

■ What is the output of the following program:

```
1 #include <stdio.h>
2
3 void main() {
4     int x = 8;
5     if (x)
6         printf("condition is true\n");
7         printf("this statement\n");
8 }
```

■ What is the output of the following program:



■ What is the output of the following program:



■ What is the output of the following program:

■ What is the output of the following program:



■ What is the output of the following program:



■ What is the output of the following program:

```
#include <stdio.h>
1
2
3
   void main() {
4
       int x = 8;
5
       if (x == 8)
6
            printf("condition is true\n");
7
       else
8
            printf("condition is false\n");
9
```

■ What is the output of the following program:



■ What is the output of the following program:

```
#include <stdio.h>
 1
 2
 3
    void main() {
        int x = 8;
 4
 5
         if (x == 8) {
 6
             printf("condition is true\n");
 7
             printf("x equals 8\n");
 8
 9
         else {
             printf("condition is false\n");
10
             printf("x does not equal 8\n");
11
12
13
     }
```

■ What is the output of the following program:



■ What is the output of the following program:

```
#include <stdio.h>
 1
 2
 3
    void main() {
        int x = 8;
 4
        if (x == 8) {
 5
 6
             printf("condition is true\n");
            printf("x equals 8\n");
 7
 8
 9
        else
10
             printf("condition is false\n");
             printf("x does not equal 8\n");
11
12
```

■ What is the output of the following program:



■ What is the output of the following program:



■ What is the output of the following program:

```
#include <stdio.h>
 1
 2
 3
    void main() {
 4
        int x = 8;
 5
        if (x == 8)
 6
             printf("condition is true\n");
 7
             printf("x equals 8\n");
 8
        else {
 9
             printf("condition is false\n");
             printf("x does not equal 8\n");
10
11
12
```

■ What is the output of the following program:

```
#include <stdio.h>
 1
 2
 3
    void main() {
 4
        int x = 8;
        if (x == 8)
 5
            printf("condition is true\n");
 6
 7
            printf("x equals 8\n");
 8
        else {
 9
            printf("condition is false\n");
            printf("x does not equal 8\n");
10
11
12
```

Nested if Statements

- Let's write a program that asks the user to enter the number of the day of the week (an integer between 1 and 7) and prints the corresponding name of the day of the week, starting with Sunday.
- This means 1 = Sunday, 2 = Monday, etc.

Nested if Statements

```
#include <stdio.h>
         1
         2
         3
            void main() {
                printf("Enter the number of the day of the week (between 1 and 7): ");
         4
         5
                int day;
         6
                scanf("%d", &day);
         7
         8
                if (day == 1)
         9
                     printf("Today is Sunday.\n");
                else if (dav == 2)
       10
       11
                     printf("Today is Monday.\n");
       12
                else if (day == 3)
       13
                     printf("Today is Tuesday.\n");
       14
                else if (day == 4)
       15
                     printf("Today is Wednesday.\n");
       16
                else if (day == 5)
       17
                     printf("Today is Thursday.\n");
                else if (day == 6)
       18
       19
                     printf("Today is Friday.\n");
       20
                else if (day == 7)
       21
                     printf("Today is Saturday.\n");
       22
                else
       23
                     printf("Error - the number you entered is invalid.");
DENTS-HUB.com
                      Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022
```

SI

Nested *if* Statements

- When we have multiple alternatives based on a single value of an integer or a character, we can replace if statements with a switch statement.
- Switch statements do not work with floats, doubles, or strings.

switch (integer or character) {

case $constant_1$:

statement(s)

break;

case constant₂:

statement(s)

break;

•••

S1

default:

statement(s)

break;

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

switch (integer or character) {

case constant₁: \blacktriangleleft

statement(s)

break;

case constant₂:

statement(s)

break;

• • •

S1

default:

statement(s)

break;

if the integer or character equals this value the following statements will be executed

DENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

switch (integer or character) {

case constant₁: \blacktriangleleft

statement(s)

if the integer or character equals this value the following statements will be executed

case constant₂:

statement(s)

break;

•••

S1

default:

statement(s)

break;

IS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

switch (integer or character) {

S1

if the integer or character equals this value case constant₁: • the following statements will be executed statement(s) break; Break causes the program to exit the switch statement case constant₂: statement(s) break; . . . default: default is an optional section it gets executed if all the 'cases' fail statement(s) break;

ENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Example – day of the week

■ Let's rewrite the previous example using a switch statement instead.

Example – day of the week

```
#include <stdio.h>
 1
 2
 3
    void main() {
        printf("Enter the number of the day of the week (between 1 and 7): ");
 4
 5
        int day;
 6
         scanf("%d", &day);
 7
8
         switch (day) {
 9
             case 1:
10
                 printf("Today is Sunday.\n");
11
                 break;
12
             case 2:
13
                 printf("Today is Monday.\n");
14
                 break:
15
             case 3:
16
                 printf("Today is Tuesday.\n");
17
                 break;
```

Example – day of the week

```
18
             case 4:
                 printf("Today is Wednesday.\n");
19
20
                 break;
21
             case 5:
22
                 printf("Today is Thursday.\n");
23
                 break;
24
             case 6:
25
                 printf("Today is Friday.\n");
26
                 break;
27
             case 7:
28
                 printf("Today is Saturday.\n");
29
                 break;
30
             default:
31
                 printf("Error - the number you entered is invalid.");
32
                 break:
33
         }
34
     }
```

break Statement

- What would happen if we forgot the break statements in the previous program?
- Let's suppose we input 1.

break Statement

```
#include <stdio.h>
 1
 2
 3
    void main() {
        printf("Enter the number of the day of the week (between 1 and 7): ");
 4
 5
        int day;
        scanf("%d", &day);
 6
 7
 8
         switch (day) {
 9
             case 1:
10
                 printf("Today is Sunday.\n");
11
             case 2:
                 printf("Today is Monday.\n");
12
                                                           the control structure.
13
             case 3:
14
                 printf("Today is Tuesday.\n");
15
             case 4:
                 printf("Today is Wednesday.\n");
16
             case 5:
17
                                                           control structure.
                 printf("Today is Thursday.\n");
18
19
             case 6:
20
                 printf("Today is Friday.\n");
21
             case 7:
22
                 printf("Today is Saturday.\n");
23
             default:
                 printf("Error - the number you entered is invalid.");
24
25
26
```

After the statement of case 1 gets executed, there's nothing to tell the program it needs to exit

So, it will continue to execute every statement is finds until it meets a break, or the end of the

DENTS-HUB.com Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022

Why Use switch Statements?

- The code we wrote using the nested if statements was 24 lines.
- The code we wrote using the switch statements was 34 lines.
- Why do we use switch statements then?

Why Use switch Statements?

- The code we wrote using the nested if statements was 24 lines.
- The code we wrote using the switch statements was 34 lines.
- Why do we use switch statements then?
- switch statements are:
 - 1. Easier to read.
 - 2. Easier to extend (add new cases).

Practical Use of switch Statements – Menus

- switch statements are often used when creating menus for you program.
- For example, you want to create a calculator that can do all five of basic arithmetic operations: +, -, *, \, %.
- And you want to let the user select which operations to do.
- Print out a menu that assigns each operations to a number and ask the user to enter the number of the operation they want to do.
- Use a switch statement to call the appropriate function.
```
#include <stdio.h>
 1
 2
 3
     void main()
 4
     {
 5
6
         printf("Menu\n");
         printf("1. Addition\n");
 7
         printf("2. Subtraction\n");
 8
         printf("3. Multiplication\n");
 9
         printf("4. Division\n");
10
11
         printf("Please select the number of the arithmetic opeeration you want to do: ");
12
         int choice;
13
         scanf("%d", &choice);
14
15
         double num_1, num_2;
16
         printf("Please enter the first number: ");
17
         scanf("%lf", &num 1);
         printf("Please enter the second number: ");
18
         scanf("%lf", &num_2);
19
20
```

```
double result;
             switch(choice) {
                 case 1:
                      result = num 1 + num 2;
                      printf("%f + %f = %f", num_1, num_2, result);
                      break:
                 case 2:
                      result = num_1 - num_2;
                      printf("%f - %f = %f", num 1, num 2, result);
                      break;
                 case 3:
                      result = num_1 * num_2;
                      printf("%f x %f = %f", num_1, num_2, result);
                      break;
                 case 4:
                      result = num_1 / num_2;
                      printf("%f / %f = %f", num_1, num_2, result);
                      break;
                 default:
                      printf("The number you entered is not valid.");
DENTS-HUB.com
                    Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022
```

Uploaded By: anonymous

ST

- Let's modify the previous code to include the modulo operation.
- Remember that modulo can only be performed on integer numbers.

#include <stdio.h>

```
void main()
{
```

```
printf("Menu\n");
printf("1. Addition\n");
printf("2. Subtraction\n");
printf("3. Multiplication\n");
printf("4. Division\n");
printf("5. Modulo\n");
```

```
printf("Please select the number of the arithmetic opeeration you want to do: ");
int choice;
scanf("%d", &choice);
```

```
if (choice == 5) {
    int num_1, num_2;
    printf("Please enter the first number: ");
    scanf("%d", &num_1);
    printf("Please enter the second number: ");
    scanf("%d", &num_2);
```

```
int result = num_1 % num_2;
printf("%d %% %d = %d", num_1, num_2, result);
```

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

```
26
             else {
                 double num_1, num_2;
    27
                 printf("Please enter the first number: ");
    28
    29
                 scanf("%lf", &num_1);
    30
                 printf("Please enter the second number: ");
    31
                 scanf("%lf", &num_2);
    32
    33
                 double result;
                 switch(choice) {
    34
    35
                     case 1:
    36
                          result = num 1 + num 2;
                         printf("%f + %f = %f", num_1, num_2, result);
    37
    38
                         break;
    39
    40
                     case 2:
                          result = num_1 - num_2;
    41
                         printf("%f - %f = %f", num 1, num 2, result);
    42
    43
                         break:
    44
                     case 3:
    45
                          result = num 1 * num 2;
    46
                         printf("%f x %f = %f", num_1, num_2, result);
    47
                         break;
    48
                     case 4:
    49
                          result = num_1 / num_2;
                         printf("%f / %f = %f", num_1, num_2, result);
    50
    51
                         break;
    52
                     default:
                         printf("The number you entered is not valid.");
    53
    54
```

Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022

Uploaded By: anonymous

S1

B.com

Example – Even or Odd

Write a program that reads a number from the user and prints a message saying if the number is even or odd.

Example – Even or Odd

```
#include <stdio.h>
 1
 2
 3
    void main()
 4
     ł
 5
         printf("Please enter an integer numbers:\n");
 6
         int a;
 7
         scanf("%d", &a);
8
 9
         if (a % 2) {
             printf("The number is odd");
10
11
         } else {
             printf("The number is even");
12
13
14
     }
```

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Example – Sides of a Triangle

- Write a program that checks if three entered integers can form a triangle.
- In a triangle, the sum of lengths of two of its sides must always be larger than the length of the third side.
 - If the triangle has sides a, b, and c, then the following statements must always be true:
 - a + b > c
 - a + c > b
 - b+c>a

Example – Sides of a Triangle

```
#include <stdio.h>
 1
 2
 3
    void main() {
 4
         printf("Please enter three integer numbers:\n");
 5
        int a, b, c;
         scanf("%d%d%d", &a, &b, &c);
 6
 7
8
         if (a + b > c \& a + c > b \& b + c > a) {
 9
             printf("These numbers can form a triangle.");
10
         } else {
11
             printf("These numbers cannot form a triangle.");
12
         }
13
     }
```

Example – Maximum of Three Number

Write a program that reads three numbers from the user and prints the maximum number.

Example – Maximum of Three Number

```
#include <stdio.h>
 1
 2
 3
    void main()
 4
     {
 5
         printf("Please enter three numbers:\n");
 6
         double a, b, c;
 7
         scanf("%lf%lf%lf", &a, &b, &c);
 8
 9
         double max = a;
         if (max < b)
10
11
             max = b;
12
         if (max < c)
13
14
             max = c;
15
         printf("The maximum number is %f", max);
16
17
```

Example – English Alphabet

- Write a program to read a character from the user and check if it is an English letter.
- We need to check if the letter is in the range of A Z or if it is in the range of a z.
- Since characters are stored in the computer as numbers, we can use relational operators with them.

Example – English Alphabet

```
#include <stdio.h>
 1
 2
 3
     void main() {
 4
         printf("Please enter a character:\n");
 5
         char c;
 6
         scanf("%c", &c);
 7
 8
         if ((c \ge 'A' \&\& c \le 'Z') || (c \ge 'a' \&\& c \le 'z')) \{
 9
             printf("This character is an English alphabet.");
         } else {
10
11
             printf("This character is an English alphabet.");
12
         }
13
     }
```

Example – Day of the Year

Write a program that takes a positive integer in the range 1 to 365 as input and outputs the day of the week. Assume that day 1 is Sunday.

Example – Day of the Year

```
#include <stdio.h>
 1
 2
 3
    void main() {
         printf("Please enter the day of the year:\n");
 4
 5
         int day;
 6
         scanf("%d", &day);
 7
 8
         int day_of_week = day % 7;
 9
10
         switch (day) {
11
             case 0:
12
                 printf("The day is Sunday.\n");
13
                 break;
14
             case 1:
15
                 printf("The day is Monday.\n");
16
                 break;
```

Example – Day of the Year



Uploaded By: anonymous

S

Example – Rounding Using if

- Remember the rounding formula we saw in the previous chapter that used the floor function? Let's try to replicate what it does using if statements.
- What we want to do:
 - Try to isolate the digit after the one we are rounding.
 - Determine if this digit is more or equal to five or if it is less than 5.
 - If it is more or equal to five, we add one to the digit before it.
 - If it is less than five, we don't add anything to the digit before.

Example – Rounding Using if

```
#include <stdio.h>
     #include <stdlib.h>
 2
 3
 4
     void main() {
 5
         printf("Please enter a number:");
         double number;
 6
         scanf("%lf", &number);
 7
 8
 9
         //To round to the third digit after the floating point:
10
         //isolate the number in the fourth digit after floating point
         int temp_number = number * 10000; //one more zero than the one we want to round
11
         int digit = temp number % 10;
12
13
14
         temp number = temp number / 10; //remove hte digit from the number
15
         //check if the digit is more than or equal to 5:
16
17
         if(digit >= 5) {
18
             //add 1 to the number:
19
             temp number += 1;
20
         }
21
         //if it is not, we don't need to do anything.
22
23
         //get the number back to its original value:
         number = temp_number / 1000.0;
24
25
26
         printf("The number rounded to the third digit is %lf", number);
27
```

S