# Branches

Department of Information System
SoICT, HUST

# `if statement`

if ( *expression* )
    *statement*

- Determines whether a statement or block is executed.
- Implements the selection instructions within an algorithm.
- Decides what to do by evaluating a Boolean expression.
- If the expression is true (non-zero), the statement or block is executed.

# What is a statement?

- Statements are lines of instructions in our programs ending with a semicolon (;).
- A compound statement or block is a series of statements surrounded by braces.

***Example***

```
{
    number = number + 1;
    printf("%d\n", number);
}
```

- An empty statement is a single semicolon.

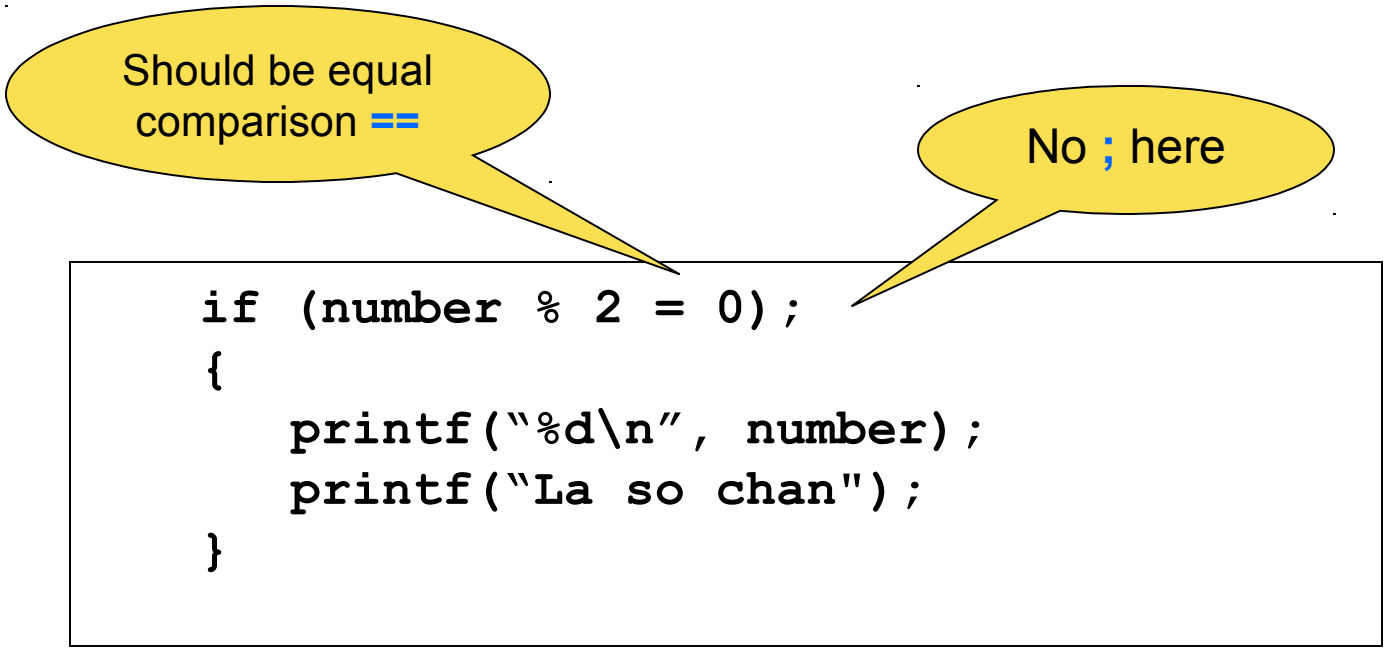
# Example

- Read in a number, and echo it if it is odd.

```
#include <stdio.h>
int main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    if (number % 2 != 0)
       printf("%d is an odd number", number);

    return 0;
}
```

there is no **then** here

# Common errors




; creates an empty statement after **if**

# else statement

if ( *expression* )

*statement1*

else *statement2*

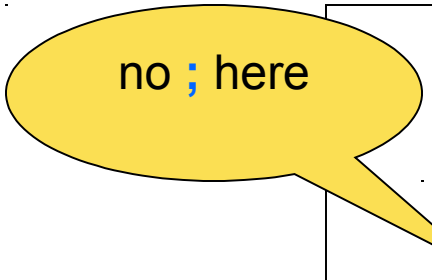
- *else* statement can only occur after an if statement
- *else* statement is only executed when the if block does not execute

# Example

- Check whether an interger is odd or even

```
#include <stdio.h>

int main()
{
  int number;

  printf(“Enter an integer: ");
  scanf("%d", &number);

  if (number % 2 != 0)
     printf("%d is an odd number\n", number);
  else
     printf("%d is an even number\n", number);

  return 0
}
```

# Common errors




```
if (number % 2 != 0)
{
   printf("%d\n", number);
   printf("is an odd number");
};
else
{
   printf("%d\n", number);
   printf("is an even number");
}
```

# Cascading if (`else-if`)

```
if (expr1)
    statement1
else if (expr2)
    statement2
else if (expr3)
    statement3
else
    statement4
```

***Example***

```
if (ch >= 'a' && ch <= 'z')
{
   printf("%c is a lowercase", ch);
}
else if (ch >= 'A' && ch <= 'Z')
{
   printf("%c is a upper case". ch);
}
else if (ch >= '0' && ch <= '9')
{
   printf("%c is a number", ch);
}
```

- Cascading **if**: Multiple alternative blocks but at most only one block will be executed
- Cascading **if** is used when we need to choose one among several conditions

# Exercise

1. Write a program to compute the total days of a month

- Algorithm

```
if (month in September, April, June, November) then
    output “30 days”
else if (month is February)
    output “28 or 29 days”
else   output “31 days”
```

# Exercises

2. Write a program to get three numbers from input and print out the maximum of those
3. Write a program to solve $ax^2 + bx + c = 0$
4. Write a program to get two numbers a,b from input and compute $y = 15x^2 + x + 12$, in which:

$$x = \begin{cases} \dfrac{a+b}{3} + b & if \quad a < b \\ 15{,}172 & if \quad a = b \\ \dfrac{a-b}{a^2+b^2} & if \quad a > b \end{cases}$$

# `switch` statement

```
switch (integer value)
  {
  case 1:  statement1;
           break; /* optional line */
  case 2:  statement2;
           break; /* optional line */
  ....
  default: default statement;
           break; /* optional line */
  }
```

- When a **switch** statement is encountered, the expression in the parentheses is evaluated and the program checks to see whether the result of that expression matches any of the constants labelled with case.
- If a match is made execution will start just after that case statement and will carry on until either the closing brace } is encountered or a *break* statement is found.
- Statements which follow the *default* case are executed for all cases which are not specifically listed.

# Example 1

```
printf("Yes/No (Y/N)?");
scanf("%c", &ch)
switch (ch)
{
   case 'y' :
   case 'Y' :
     printf("say yes");
     break;
   default :
     printf("say no");
}
```

# Example 2

```
switch (digit){
   case 0 : printf ("zero");
            break;
   case 1 : printf ("one");
            break;
   case 2 : printf ("two");
            break;
   ...
   case 9 : printf ("nine");
            break;
   default:
       printf ("others");
}
```

# Exercises

- Display grade of a student based on marks
- diem = 9, 10: excellent
- diem = 7, 8: good
- diem = 5, 6: average
- other: weak

# Solution

- Display grade of a student based on marks

```
switch (grade)
{
  case 9:
  case 10:
    printf("excellent \n");
    break;
  case 7:
  case 8:
    printf("good \n");
    break;
  case 5:
  case 6:
    printf("average \n");
    break;
  default:
    printf("weak \n");
}
```

```
if (grade ==9|| grade ==10)
{
   printf("excellent \n");
}
else if (grade ==7||
   grade==8)
{
   printf("good \n");
}
else if (grade==5|| grade==6)
{
   printf("average \n");
}
else
{
   printf("weak \n");
}
```

# using **break**

- When a case of the switch statement is found, statements are carried out from this point
- All following statements are carried out until a break statement
- **break** is a handy way of jumping straight out of the switch block

```
int a=1;
switch ( a ) {
    case 1:
        printf("a=1\n");
    case 2:
        printf("a=2\n");
        break;
    case 3:
        printf("a=3\n");
}
```

Output:

a=1
a=2

# Exercises

1. Write a program to get two numbers a,b from input and compute $y = 15x^2 + x + 12$, in which:

$$x = \begin{cases} \dfrac{a+b}{3} + b & if \quad a < b \\ 15{,}172 & if \quad a = b \\ \dfrac{a-b}{a^2+b^2} & if \quad a > b \end{cases}$$

2. Write a program to get an integer n ($1 \le n \le 10$) and display the English name of that number. For example, n = 2, display 2 → two.