Condition

SESSION 5

Objective

Explain the selection construct

- if statement
- if – else statement
- if – else – if statement
- Nested if statement

Switch statement

Conditional Statement

Conditional statements enable us to change the flow of the program.

A conditional statement evaluates to either true or false value

Example :

To find whether a number is even or odd we proceed as follows :

1. **Accept a number**
2. **Find the remainder by dividing the number by 2**
3. **If the remainder is zero, the number is “EVEN”**
4. **Or if the remainder is not zero the number is “ODD”**

Selection constructs

C supports two types of selection statements

The if statement

The switch statement

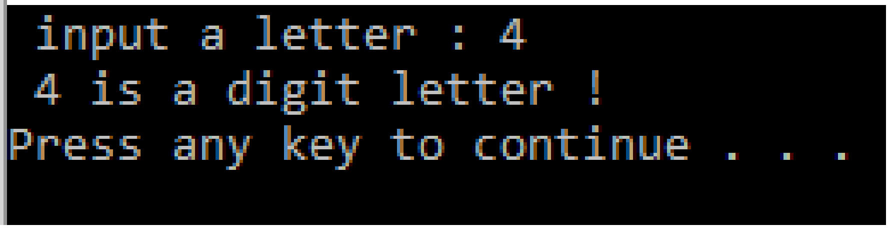
The if - statement

Syntax

```
    if (expression)   {
        statement ;
        statement ;
        …
    }
```

Example

```
char ch;
printf("input a letter : ");
scanf("%c",&ch);
if(ch>='0' & ch <='9'){
    printf("%c is a digit letter !\n",ch);
}
```



```
 input a letter : 4
 4 is a digit letter !
Press any key to continue . . .
```

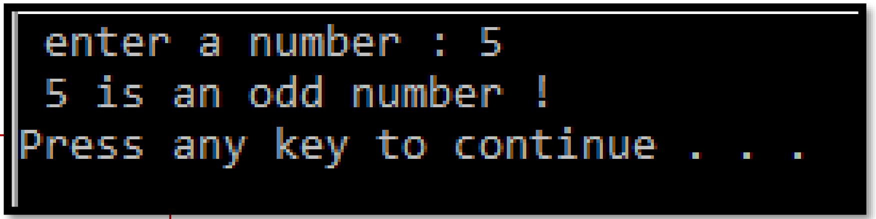
The if – else statement -1/2

Syntax

```
if (expression)  {
    statement ;
}
else{
    statement ;
}
```

Example

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



```
 enter a number : 5
 5 is an odd number !
Press any key to continue . . .
```

The if – else statement -2/2

- If the **if expression** evaluates to **true**, the block following the if statement is executed
- If the **if expression** evaluates to **false**, the statements following the else expression take over control
- The else statement is optional.

```
if (expression)  {
    statements ;
}
else{
    statements ;
}
```

The if–else–if statement-1/2

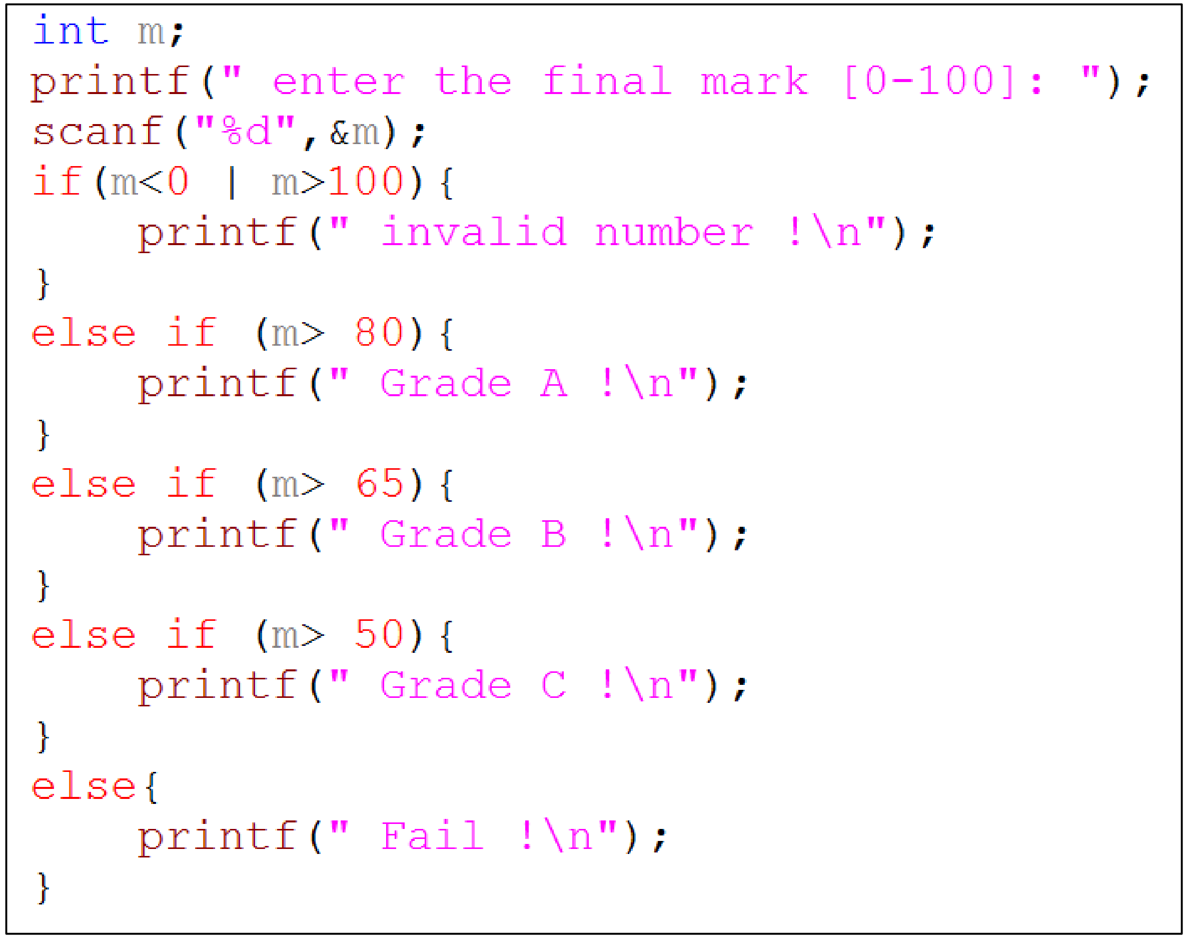
Syntax

```
if (expression)  {
    statements;
}
else if (expression){
    statements;
}
else if (expression){
    statements;
}
else{
    statements;
}
```

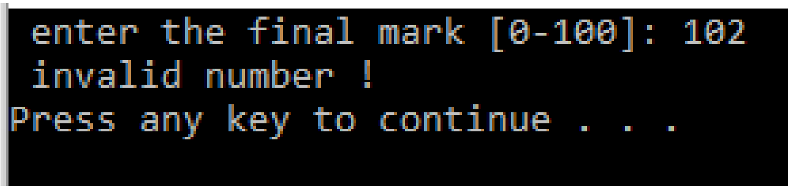
- The if – else – if statement is also known as the if-else-if ladder or the if-else-if staircase
- The conditions are evaluated from the top downwards

The if – else – if statement-2/2

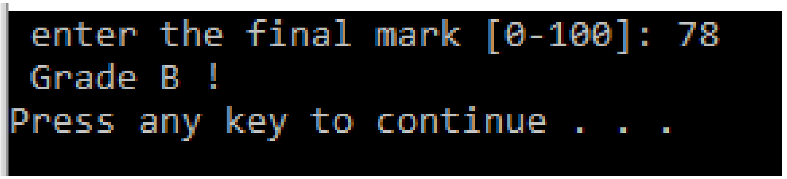
Example



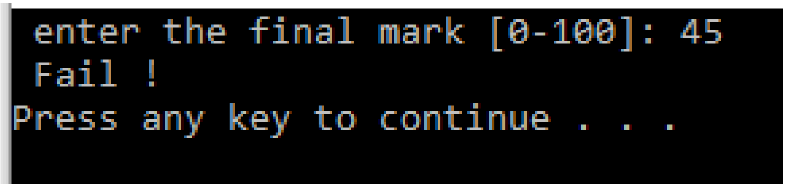
```
int m;
printf(" enter the final mark [0-100]: ");
scanf("%d",&m);
if(m<0 | m>100){
    printf(" invalid number !\n");
}
else if (m> 80){
    printf(" Grade A !\n");
}
else if (m> 65){
    printf(" Grade B !\n");
}
else if (m> 50){
    printf(" Grade C !\n");
}
else{
    printf(" Fail !\n");
}
```



```
 enter the final mark [0-100]: 102
 invalid number !
Press any key to continue . . .
```



```
 enter the final mark [0-100]: 78
 Grade B !
Press any key to continue . . .
```



```
 enter the final mark [0-100]: 45
 Fail !
Press any key to continue . . .
```

Nested if - 1 /2

Syntax

```
if (exp 1)  {
      if (exp 2){
          statements;
      }
      if (exp 3){
          statements;
      }
      else statement /*with if (exp3)*/
}
else{
  statements;
}
```

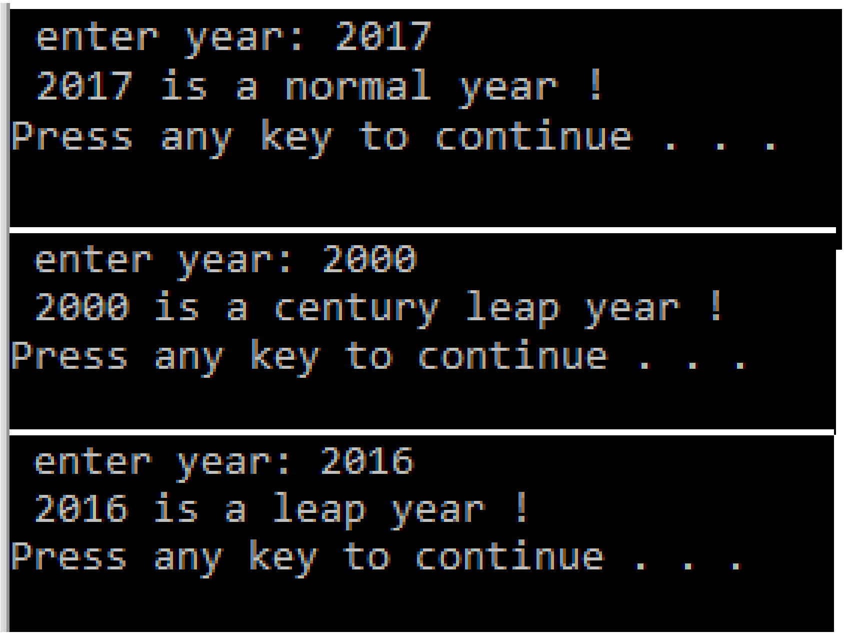
- The nested **if** is an **if** statement, which is placed within another **if** or **else**

- In C, an **else** statement always refers to the nearest **if** statement that is within the same block and is not already associated with an other **if**

According to ANSI standards, a compiler should support at least 15 levels of nesting

Nested if – Example 2 /2

```
int y;
printf(" enter year: ");
scanf("%d",&y);
if(y%4==0){
    if(y%100!=0){
        printf(" %d is a leap year !\n", y);
    }
    else if(y%400==0){
        printf(" %d is a century leap year !\n", y);
    }
    else printf(" %d is a century year !\n", y);
}
else printf(" %d is a normal year !\n", y);
```



```
 enter year: 2017
 2017 is a normal year !
Press any key to continue . . .
```

```
 enter year: 2000
 2000 is a century leap year !
Press any key to continue . . .
```

```
 enter year: 2016
 2016 is a leap year !
Press any key to continue . . .
```

The switch statement-1/2

Syntax

```
switch (expression)  {
  case constant1:
      statements;
      break;
  case constant2:
      statements;
      break;
  case constant3:
      statements;
      break;
      …
  default:
      statements;
}
```

- The **switch** statement is a multi-way decision maker that tests the value of an expression against a list of integers or character constants

- When a match is found, the statements associated with that constant are executed

The switch statement - 2/2

Example

```
char ch;
printf ("\nEnter a lower cased alphabet (a - z) : ");
scanf("%c", &ch);
if (ch < 'a' | ch > 'z')
    printf("\nCharacter not a lower cased alphabet");
else
    switch (ch) {
        case 'a' :
        case 'e' :
        case 'i' :
        case 'o' :
        case 'u' :
            printf("\nCharacter is a vowel"); break;
        case 'z' :
            printf("\nLast Alphabet (z) was entered"); break;
        default :
            printf("\nCharacter is a consonant"); break;
    }
}
```

END