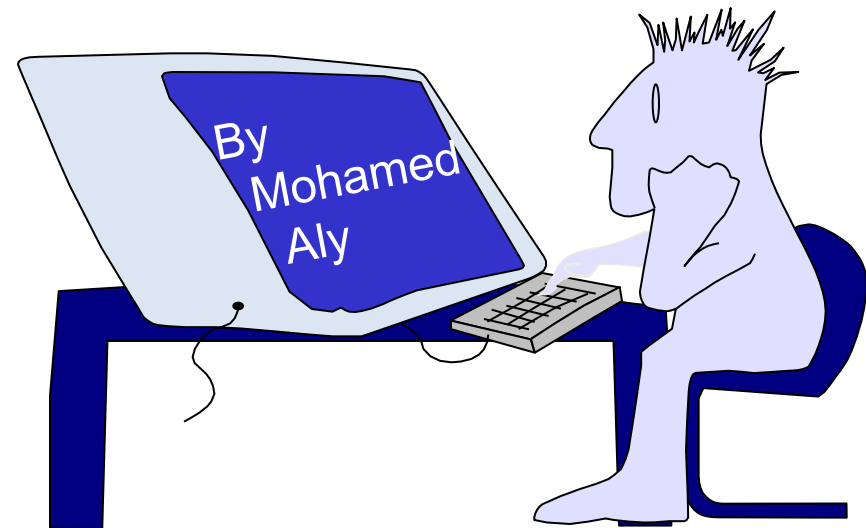


Embedded C

C Programming Part 2



Flow control statements

- **if-else**
- **switch**
- **for**
- **while**
- **do-while**
- **goto**
- **break**
- **continue**
- Compound statements can be placed anywhere a single statement can be placed
- It is a must to have at least single statement in their body, it is also possible to have no statement at all, i.e., the ***empty statement***

Flow control statements(Cont.)

if-else examples

```
if ( grade >= 50 )  
    printf( "Passed. \n");  
else  
    printf( "Failed. \n");
```

```
if ( grade >= 50 )  
    printf( "Passed. \n" );  
else  
{  
    printf( "Failed. \n" );  
    printf( "You must take this course again. \n" );  
}
```

Flow control statements(Cont.)

Nested if-else example

```
if( grade >= 85 )
    Printf( "Excellent \n" );
else
    if( grade >= 75 )
        printf( "Very Good \n" );
    else
        if( grade >= 65 )
            printf( "Good \n" );
        else
            if( grade >= 50 )
                printf( "Passed \n" );
            else
                printf( "Failed \n" );
```

```
if( grade >= 85 )
    Printf( "Excellent \n" );
else if( grade >= 75 )
    printf( "Very Good \n" );
else if( grade >= 65 )
    printf( "Good \n" );
else if( grade >= 50 )
    printf( "Passed \n" );
else
    printf( "Failed \n" );
```

Flow control statements(Cont.)

switch statement

switch statement anatomy

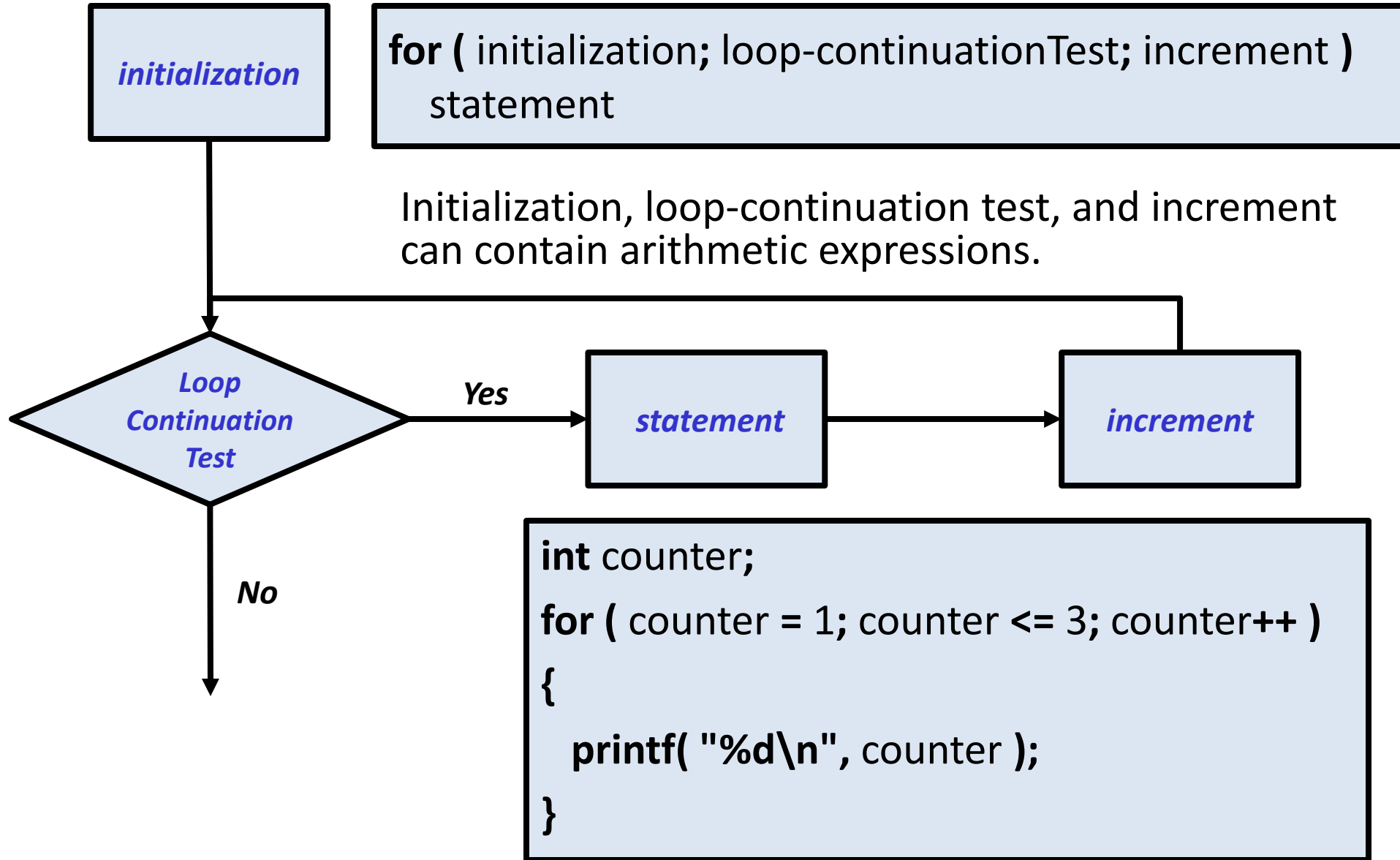
```
switch ( variable )
{
    case 'value_1':
        actions_1
    case 'value_2':
        actions_2
    default:
        default_actions
}
```

```
switch ( grade )
{
    case 'e':
        ++eCount;
        break;
```

```
    case 'v':
        ++vCount;
        break;
    case 'g':
        ++gCount;
        break;
    case "p":
        ++pCount;
        break;
    case '\n':
    case ' ':
        break;
    default:
        printf( "Incorrect letter grade entered. " );
        printf( " Enter a new grade. \n" );
        break;
}
```

Flow control statements(Cont.)

for loop



Flow control statements(Cont.)

while examples

```
int counter = 1;
while ( counter <= 3 )
{
    printf( "%d\n", counter );
    ++counter;
}
counter = 0;
while ( ++counter <= 3 )
    printf( "%d\n", counter );
```

```
int counter = 1;
int result  =1;
while ( counter <= 3 )
{
    result  = result * counter;
    counter = counter + 1;
}
```

Flow control statements(Cont.)

do-while example

```
int counter = 0;
do
{
    printf( "%d\n", counter );
}
while (++counter <= 3);
```


Flow control statements(Cont.)

goto statement

- Unconditional branching to first statement after specified label
- A label is an identifier followed by a colon

```
label:  
/*Statement;*/  
/*Compound statement*/  
/* ..... */  
/* ..... */  
/* ..... */  
/*Statement;*/  
goto label;
```

Flow control statements(Cont.)

break and continue statements

- **break**
 - Causes immediate exit from a **while**, **for**, **do...while** or **switch** statement
 - Program execution continues with the first statement after the statement
- **continue**
 - Skips the remaining statements in the body of a **while**, **for**, or **do...while** statement in other words proceeds with the next iteration of the loop
 - For **while** and **do...while**, Loop-continuation test is evaluated immediately after the **continue** statement is executed
 - For **for**, Increment expression is executed, then the loop-continuation test is evaluated

Empty statement

- The empty statement is represented by placing a semicolon (;) where a statement would normally be

Thanks

Embedded C