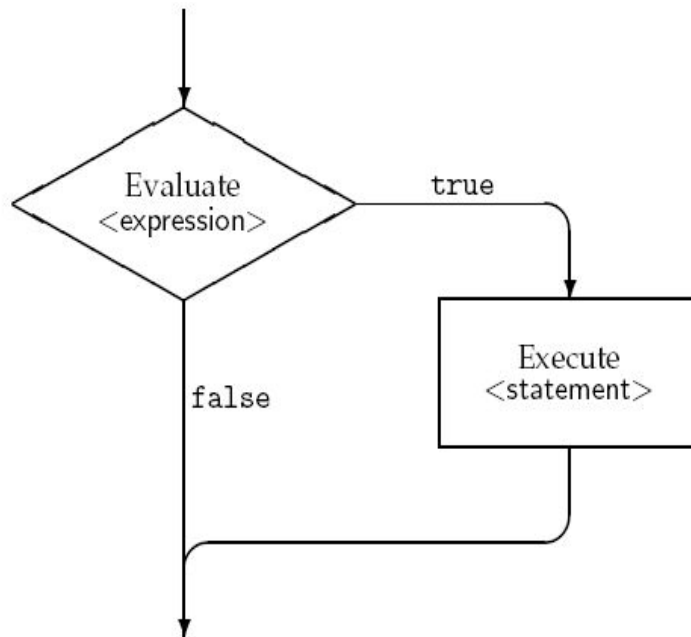


# Introduction to C Programming

Section 3

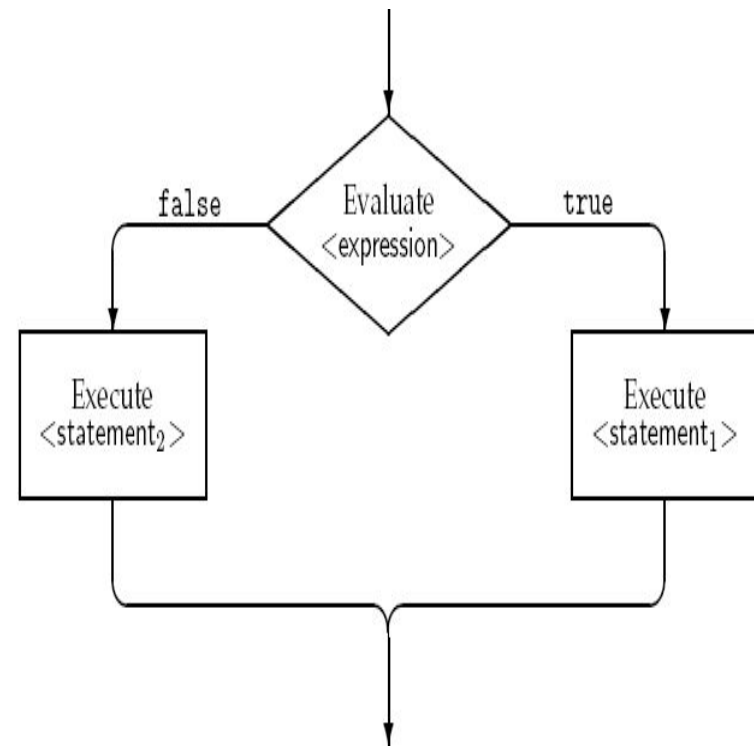
# Flowchart

if(<expression>)  
    <statement>



if(<expression>)  
    <statement1>

else  
    <statement2>



# Program to check if number is even or odd

```
#include <stdio.h>

int main(void){

    int number_to_test, remainder;
    printf("Enter your number to be tested: ");
    scanf("%d", &number_to_test);
    remainder = number_to_test % 2;
    if(remainder == 0)
        printf ("The number is even.\n")
    else
        printf ("The number is odd.\n")

    return 0;
}
```

# Map numeric grade to alphabetic

```
int numg;  
char alphag;  
if(numg < 25)  
    alphag = 'D';  
if((numg >= 25) && (numg < 50))  
    alphag = 'C';  
if((numg >= 50) && (numg < 75))  
    alphag = 'B';  
if(numg >= 75)  
    alphag = 'A';
```

# More than two choices

- To avoid repeating conditions in if statements
- To avoid running unnecessary statements
- **Nested** if: check multiple conditions
  - <Statements 1> becomes an if-else statement
  - <Statements 2> becomes an if-else statement
  - Repeat it as many as needed

# Nested if

```
if(<condition 1>
    <statement 1>
else{
    if(<condition 2>)
        <statement 2>
    else
        <statement 3>
}
```

```
if(<condition 1>
    <statement 1>
else if(<condition 2>)
    <statement 2>
else
    <statement 3>
```

# Map numeric grade to alphabetic

```
int numg;  
char alphag;  
if(numg < 50){  
    if(numg < 25)  
        alphag = 'D';  
    else  
        alphag = 'C';  
}  
else{  
    if(numg < 75)  
        alphag = 'B';  
    else  
        alphag = 'A';  
}
```

# Map numeric grade to alphabetic

```
int numg;  
char alphag;  
if(numg < 25)  
    alphag = 'D' ;  
else{  
    if(numg < 50)  
        alphag = 'C' ;  
    else{  
        if(numg < 75)  
            alphag = 'B' ;  
        else  
            alphag= 'A' ;  
    }  
}
```



# switch-case: Multiple choices

- Multiple conditions
  - If-else if-else if-....
- Select from alternative values of a variable
  - switch-case
  - Values should be **constant** not expression: **i**, **i+j**,
- Values & Variables should be **int** or **char**

```
switch(variable){  
  
    case value1:  
        <statements 1>  
    case value2:  
        <statements 2>  
  
}
```

# How does switch-case work?

- Each switch-case can be rewritten by If-else
  - if-else version of switch-case in the previous slide

```
if(variable == value1) {  
    <statements 1>  
    <statements 2>  
}  
else if(variable == value2) {  
    <statements 2>  
}
```

# switch-case: complete version

```
switch(variable){  
    case value1:  
        <statements 1>  
        break;  
    case value2:  
        <statements 2>  
        break;  
    default:  
        <statements 3>  
}
```

```
If(variable == value1)  
{  
    <statements 1>  
}  
else if(variable == value2)  
{  
    <statements 2>  
}  
else{  
    <statements 3>  
}
```

# Simple Calculator

```
#include <stdio.h>
#include <stdlib.h>
int main(void){
    int res, opd1, opd2;
    char opr;
    printf("Operand1 : ");
    scanf("%d", &opd1);
    printf("Operand2 : ");
    scanf("%d", &opd2);
    printf("Operator : ");
    scanf(" %c", &opr);
```

```
    switch(opr){
        case '+':
            res = opd1 + opd2;
            break;
        case '-':
            res = opd1 - opd2;
            break;
        case '/':
            res = opd1 / opd2;
            break;
```

# Simple Calculator cont.

```
case '*':
    res = opd1 * opd2;
    break;
default:
    printf("Invalid operator \n");
    return -1;
}
printf("%d %c %d = %d\n", opd1, opr, opd2, res);
return 0;
}
```

# switch-case

- All values used in case should be different

```
switch(i) { //Error
```

```
    case 1:
```

```
    ...
```

```
    case 2:
```

```
    ...
```

```
    case 1:
```

```
    ...
```

# switch-case

- All values must be value, not expression of variables

```
switch(i) { //Error
```

```
case j:
```

```
...
```

```
case 2:
```

```
...
```

```
case k+10:
```

```
...
```