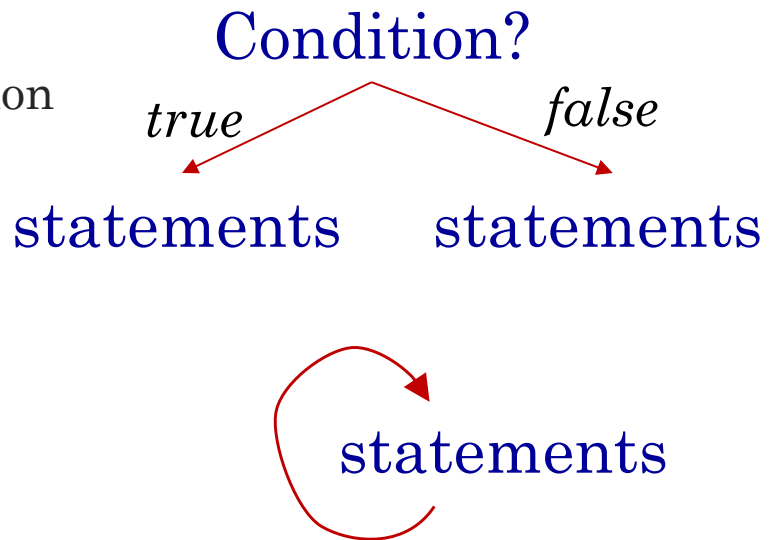


CSE 220 – C Programming

Selection Statements

Control of Flow

- Selection statements:
 - Select a particular path of execution
- Iteration statements:
 - Repeat a particular fragment
- Jump statements:
 - Jump to another place in the code



Logical Expressions

- Relation operators: <, <=, >=, >
- Equality operators: ==, !=
- Logical operators: !, &&, ||
- Produce 0 or 1

`(1 <= j) && (j <= 100)`

`(i < j) == (j < k)`

`!answer || count >= 5`

Logical Expressions

- $!(a \mid \mid b) == !a \&\& !b$
- $!(a \&\& b) == !a \mid \mid !b$

`!answer || count >= 5`

`!(answer && count < 5)`

If Statement

`if (expression) statement`

- `()` required around expression
- `if`: lowercase
- If expression evaluates to non zero, then statement is executed

```
int answer;  
printf("Guess my lucky number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
printf("You are correct!\n");  
printf("We are done");
```

If Statement

```
if ( expression ) statement
```

- () required around expression
- if: lowercase
- If expression evaluates to non zero, then statement is executed

```
int answer;  
printf("Guess my lucky number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
    printf("You are correct!\n");  
printf("We are done");
```

If Statement

```
if ( expression ) statement
```

- () required around expression
- if: lowercase
- If expression evaluates to non zero, then statement is executed

```
int answer;  
printf("Guess my lucky number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
    printf("You are correct!\n");  
printf("We are done");
```

If Statement

`if (expression) statement`

- Indentation only affects readability, not code execution

```
int answer;  
printf("Guess my lucky number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
    printf("You are correct!\n");  
    printf("We are done");
```

Executed only
if answer is 5

Always
executed

Compound Statement

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

- Use `{ }` to execute multiple statements
- Compound statement can be written on one line

```
int answer;  
printf("Guess my lucky number:\n");  
scanf("%d", &answer);  
if (answer == 5) {  
    printf("You are correct!\n");  
    printf("How did you know?\n");  
}  
printf("We are done");
```

Compound Statement

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

- Another coding style:

```
int answer;  
printf("Guess my lucky number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
{  
    printf("You are correct!\n");  
    printf("How did you know?\n");  
}  
printf("We are done");
```

Else Clause

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

- Else statements are executed if expression evaluates to 0

```
int answer;  
printf("Guess my lucky  
number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
    printf("Correct!\n");  
if (answer != 5)  
    printf("Wrong!\n");
```

```
int answer;  
printf("Guess my lucky  
number:\n");  
scanf("%d", &answer);  
if (answer == 5)  
    printf("Correct!\n");  
else  
    printf("Wrong!\n");
```

Cascading if statements

```
if (delta < 0)
    printf("No real roots\n");
else
    if (delta == 0)
        printf("One real root\n");
    else
        printf("Two roots\n");
```

```
if (delta < 0)
    printf("No real roots\n");
else if (delta == 0)
    printf("One real root\n");
else
    printf("Two roots\n");
```

Use when:

- series of conditions
- stop as soon as one if true

Cascading if statements

```
if (age < 40)
    printf("You are young\n");
else if (age < 50)
    printf("You are in your forties\n");
else if (age < 60)
    printf("You are in your fifties\n");
else if (age < 70)
    printf("You are in your sixties\n");
else if (age < 80)
    printf("You are in your seventies\n");
else if (age < 90)
    printf("You are in your eighties\n");
else if (age < 100)
    printf("You are in your nineties\n");
else
    printf("Really?\n");
```

```
if (age < 40)
    printf("You are young\n");
else
    if (age < 50)
        printf("In your 40s\n");
    else
        if (age < 60)
            printf("In your 50s\n");
        else
            if (age < 70)
                printf("In your 60s\n");
            else
                if (age < 80)
                    printf("In your 70s\n");
                else
                    if (age < 90)
                        printf("In your 80s\n");
                    else
                        if (age < 100)
                            printf("In your 90s\n");
                        else
                            printf("Really?\n");
```

Best-Form Cascading if statements

```
if (age < 40) {  
    printf("You are young\n");  
} else { if (age < 50) {  
    printf("You are in your forties\n");  
} else { if (age < 60) {  
    printf("You are in your fifties\n");  
} else { if (age < 70)  
    printf("You are in your sixties\n");  
} else { if (age < 80)  
    printf("You are in your seventies\n");  
} else { if (age < 90)  
    printf("You are in your eighties\n");  
} else { if (age < 100)  
    printf("You are in your nineties\n");  
} else {  
    printf("Really?\n");  
}}}}}}}
```

Example

```
int a, b, c, z;
printf("Enter 3 integers:\n");
scanf("%d%d%d", &a, &b, &c);
if (a < b)
    if (b < c)
        z = c;
    else
        z = b;
else
    if (a > c)
        z = a;
    else
        z = c;
```

Example

```
int a, b, c, z;  
printf("Enter 3 integers:\n");  
scanf("%d%d%d", &a, &b, &c);  
if (a > b && a > c)  
    z = a;  
else if (b > c)  
    z = b;  
else  
    z = c;
```


Dangling else problem

```
if ( salary < 200000)
    if (salary > 50000)
        printf("You are doing alright\n");
else
    printf("You make a lot of money");
```

- What is the output if salary is

\$100,000 \$250,000 \$10,000?

Dangling else problem

```
if ( salary < 200000)
    if (salary > 50000)
        printf("You are doing alright\n");
    else
        printf("You make a lot of money");
```

- Else is matched to the nearest if
- Use {} even with one statement only

Exercise

What is the output?

```
int j = 30;  
if (10 < j < 20) {  
    printf("%d is between 10 and 20\n", j);  
}
```

- Nothing
- 30 is between 10 and 20
- 30 is **not** between 10 and 20
- Something Else

Exercise

What is the output?

```
int j = 30;  
if (10 < j < 20) {  
    printf("%d is between 10 and 20\n", j);  
}
```

Evaluate $10 < j < 20$:

$((10 < j) < 20)$

$((10 < 30) < 20)$

$(1 < 20)$

1

Conditional Expressions

`expr1 ? expr2 : expr3`

- Conditional operator: ? and :
- Ternary operator

```
int k = a > b ? a : b;
```

```
int k;  
if (a > b) {  
    k = a;  
} else {  
    k = b;  
}
```

Conditional Expressions

`expr1 ? expr2 : expr3`

- Conditional operator: ? and :
- Ternary operator

```
float newValue = op == '*' ? value*2 : value/2;
```

```
float newValue;  
if (op == '*') {  
    newValue = value*2;  
} else {  
    newValue = value/2;  
}
```

Conditional Expressions

```
int k = ( i > 0 ? 1 : -1 ) * i;
```

- Make programs shorter but harder to read
- Use with simple expressions

```
int k;  
if i > 0  
    k = i  
else  
    k = -i;
```

```
int k = i > 0 ? i : -i
```

Switch Statement

```
switch (expression) {  
    case constant-expr : statements;  
    .....  
    case constant-expr : statements;  
    default: statements;  
}
```

- Use to compare an expression with a number of values

Switch Statement

```
switch (day) {  
    case 1: printf("Monday\n");  
            break;  
    case 2: printf("Tuesday\n");  
            break;  
    case 3: printf("Wednesday\n");  
            break;  
    case 4: printf("Thursday\n");  
            break;  
    case 5: printf("Friday\n");  
            break;  
    case 6: printf("Saturday\n");  
            break;  
    case 7: printf("Sunday\n");  
            break;  
    default: printf("??? \n");  
            break;  
}
```

```
if (day == 1) {  
    printf("Monday\n");  
} else if (day == 2) {  
    printf("Tuesday\n");  
} else if (day == 3) {  
    printf("Wednesday\n");  
} else if (day == 4) {  
    printf("Thursday\n");  
} else if (day == 5) {  
    printf("Friday\n");  
} else if (day == 6) {  
    printf("Saturday\n");  
} else if (day == 7) {  
    printf("Sunday\n");  
} else {  
    printf("??? \n");  
}
```

Switch Statement

```
switch (day) {  
    case 1: printf("Monday\n");  
             break;  
    case 2: printf("Tuesday\n");  
             break;  
    case 3: printf("Wednesday\n");  
             break;  
    case 4: printf("Thursday\n");  
             break;  
    case 5: printf("Friday\n");  
             break;  
    case 6: printf("Saturday\n");  
             break;  
    case 7: printf("Sunday\n");  
             break;  
    default: printf("??? \n");  
             break;  
}
```

- Easier to read
- Faster

Switch Statement

```
switch (day) {  
    case 1: printf("Monday\n");  
             break;  
    case 2: printf("Tuesday\n");  
             break;  
    case 3: printf("Wednesday\n");  
             break;  
    case 4: printf("Thursday\n");  
             break;  
    case 5: printf("Friday\n");  
             break;  
    case 6: printf("Saturday\n");  
             break;  
    case 7: printf("Sunday\n");  
             break;  
    default: printf("???\n");  
             break;  
}
```

- Switch must be followed by int (or char)
- No braces after case label
- Constant expressions:
 - 1, 4+3,
 - $x + 1$: if x is defined by a constant macro: `#define x 5`
- Duplicate labels not allowed
- *break*: exit the switch statement
- *default*: executed if no match
- Can group case labels

Switch Statement

```
switch (day) {  
    case 1: case 2: case 3: case 4: case 5:  
        printf("Week day\n");  
        break;  
    case 6: case 7:  
        printf("Weekend\n");  
        break;  
    default: printf("I don't know\n");  
        break;  
}
```

Switch Statement

```
switch (day) {  
    case 1: case 2: case 3: case 4: case 5:  
        printf("Week day\t");  
    case 6: case 7:  
        printf("Weekend\t");  
    default: printf("I don't know\t");  
}
```

If day is 3, output is?

Nothing

Weekend

Week Day

I don't know

Switch Statement

```
switch (day) {  
    case 1: case 2: case 3: case 4: case 5:  
        printf("Week day\t");  
    case 6: case 7:  
        printf("Weekend\t");  
    default: printf("I don't know\t");  
}
```

If day is 6, output is?

Nothing

Weekend

Week Day

I don't know

Pitfalls

- Lazy execution:

```
i > 20 && j++ < 5
```

- Using assignment op:

```
if (i = 2) {  
    printf("Are you sure?\n");  
}
```

- Dangling else
- Forgetting break in the switch statement

Summary

- Logical Expressions
- If statement
- Conditional statement
- Switch statement