Control Instructions

IN 1101 PROGRAMMING FUNDAMENTALS

Decisions!

- We need to alter the actions based on the circumstances.
- C allows to perform different sets of actions depending on the circumstance.
- By default, the instructions of a program are executed sequentially.
- Decision control instructions can be implemented in C using:
  - The **if** statements.
  - The **if-else** statement.
  - The **conditional** operators.
  - The **Switch** statements.

The if Statement

- C uses the keyword if to implement the decision control instruction.

```
if ( this condition is true )
    execute this statement ;
```

- The keyword if tell the compiler that what follows is a decision control instruction.
- Can express a condition using C’s relational operators.

Relational Operators

Expression	Is true if
x == y	x is equal to y
x != y	x is not equal to y
x < y	x is less than y
x > y	x is greater than y
x <= y	x is less than or equal to y
x >= y	x is greater than or equal to y

Example

While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000. If quantity and price per item are input through the keyboard, write a program to calculate the total expenses.

1. Draw the appropriate flowchart for the above scenario.

2. Write the C code for the above scenario.

Example

The current year and the year in which the employee joined the organization are entered through the keyboard. If the number of years for which the employee has served the organization is greater than 3, then a bonus of Rs. 2500/- is given to the employee. If the years of service are not greater than 3, then the program should do nothing.

Write the C program for the above scenario.

The if-else Statement

- We execute one group of statements if the expression evaluates to true and another group of statements if the expression evaluates to false.

```
if ( this condition is true ) {
        execute these statements ;
}
else{
        execute these statements;
}
```

- **Nested if-else**s - we write an entire if-else construct within either the body of the if statement or the body of an else statement

Example

In a company an employee is paid as under:

If his basic salary is less than Rs. 1500, then HRA = 10% of basic salary and DA = 90% of basic salary. If his salary is either equal to or above Rs. 1500, then HRA = Rs. 500 and DA = 98% of basic salary. If the employee's salary is input through the keyboard write a program to find his gross salary.

Write the C code for the above scenario using if-else statements.

Logical Operators

- C allows usage of three logical operators,

&& - AND

|| - OR

! - NOT

```
if (condition 1 ) && (condition 2) {
        execute these statements ;
}
else{
       execute these statements;
}
```

Example

The marks obtained by a student in 5 different subjects are input through the keyboard. The student gets a division as per the following rules:

- Percentage above or equal to 60 - First division
- Percentage between 50 and 59 - Second division
- Percentage between 40 and 49 - Third division
- Percentage less than 40 - Fail

Write a program to calculate the division obtained by the student using logical operators.

The else if Clause

```
If(Condition1){
    Statement1;
}
Else if(Condition2){
    Statement2;
}
Else if(Condition3){
    Statement3;
}
Else{
    Final statement block;
}
```

Example

Rewrite the C code for above example using else if blocks.

Conditional Operators

- The conditional operators ? and : are sometimes called ternary operators.
- It form a kind of foreshortened if-then-else.

expression 1 ? expression 2 : expression 3

"if **expression 1** is true, then the value returned will be **expression 2**, otherwise the value returned will be **expression 3**".

The Switch Statements

- **switch-case-default** - a control statement that allows us to make a decision from the number of choices.

```
switch ( integer expression )
{
      case constant 1 :
            do this ;
      case constant 2 :
            do this ;
      case constant 3 :
            do this ;
      default :
            do this ;
}
```

Break Statement

- **Break** statement can be used to beak the execution.

```
switch ( integer expression )
{
        case constant 1 :
                do this ;
              break;
        case constant 2 :
                do this ;
              break
        default :
                do this ;
              // No need of a beak here
}
```

- **H/W** – Does **Switch** works faster than equivalent **if-else**?

Homework

- Find what '**goto**' keywork does in C language?
- What are the advantages of having '**goto**' in the program?
- Why you should avoid using '**goto**' in everywhere in your program?

Questions?