| | |
|---|---|
| **Soal Praktikum**<br>*Practicum Case* | |
| **COMP6360004**<br>Algorithm and Programming | BINUS UNIVERSITY |
| **Teknik Informatika**<br>*Computer Science* | **E222-COMP6360004-EP01-03** |
| **Periode Berlaku** Semester Ganjil 2022/2023<br>***Valid on*** *Odd Semester Year 2022/2023* | **Revisi 00**<br>*Revision 00* |

## Learning Outcomes

- Apply syntax and functions in C language in problem solving
- Construct a program using C language in problem solving

## Topic

- Session 03 - Program Control Selection

## Sub Topics

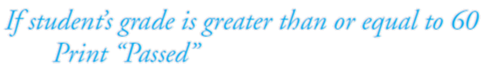
- If-Else
- Using ? :
- Nested If
- Switch-Case
- Create simple program using selection and logical operators

## Tutorial/Panduan

Normally, statements in a program are executed one after the other in the order in which they're written. This is called sequential execution. Various C statements we'll soon discuss enable you to specify that the next statement to be executed may be other than the next one in sequence. This is called transfer of control. C has only seven control statements: sequence, three types of selection and three types of repetition. Each C program is formed by combining as many of each type of control statement as is appropriate for the algorithm the program implements.

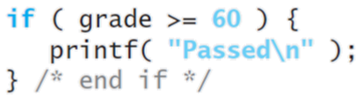
### The if Selection Statement

Selection structures are used to choose among alternative courses of action. For example, suppose the passing grade on an exam is 60. The pseudocode statement determines if the condition "student's grade is greater than or equal to 60" is true or false. If the condition is true, then "Passed" is printed, and the next pseudocode statement in order is "performed" (remember that pseudocode is not a real programming language). If the condition is false, the printing is ignored, and the next pseudocode statement in order is performed. The second line of this selection structure is indented. Such indentation is optional, but it's highly recommended as it helps emphasize the inherent structure of structured programs. The C compiler ignores white-space characters like blanks, tabs and newlines used for indentation and vertical spacing.

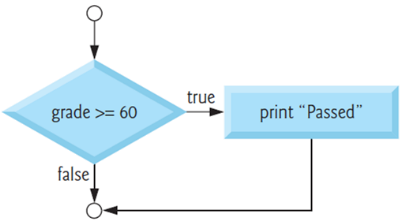


```
If student's grade is greater than or equal to 60
    Print "Passed"
```

The preceding pseudocode If statement may be written in C as this.

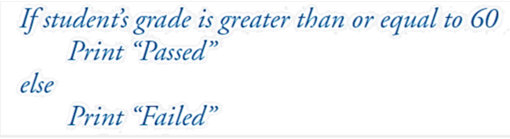


```
if ( grade >= 60 ) {
   printf( "Passed\n" );
} /* end if */
```

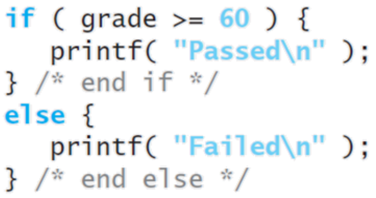


## The if…else Selection Statement

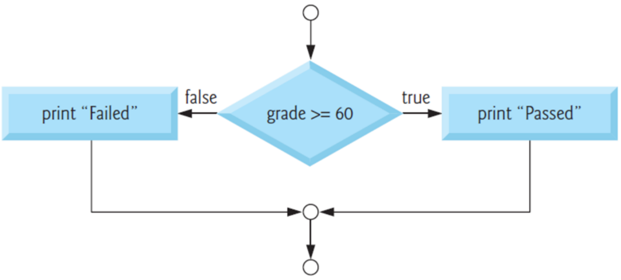
The if selection statement performs an indicated action only when the condition is true; otherwise the action is skipped. The if…else selection statement allows you to specify that different actions are to be performed when the condition is true than when the condition is false. For example, the pseudocode statement prints Passed if the student's grade is greater than or equal to 60 and prints Failed if the student's grade is less than 60. In either case, after printing occurs, the next pseudocode statement in sequence is "performed." The body of the else is also indented. Whatever indentation convention you choose should be carefully applied throughout your programs. The preceding pseudocode If statement may be written in C as this.



```
If student's grade is greater than or equal to 60
    Print "Passed"
else
    Print "Failed"
```

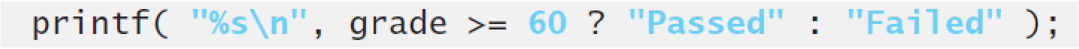


```
if ( grade >= 60 ) {
   printf( "Passed\n" );
} /* end if */
else {
   printf( "Failed\n" );
} /* end else */
```



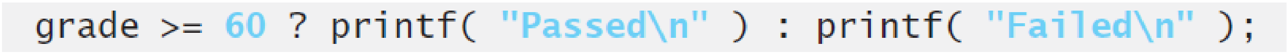
## if…else statements using conditional operator (?:)

C provides the conditional operator (?:) which is closely related to the if…else statement. The conditional operator is C's only ternary operator—it takes three operands. The operands together with the conditional operator form a conditional expression. The first operand is a condition. The second operand is the value for the entire conditional expression if the condition is true and the third operand is the value for the entire conditional expression if the condition is false. For example, the printf statement contains a conditional expression that evaluates to the string literal "Passed" if the condition grade >= 60 is true and evaluates to the string literal "Failed" if the condition is false. The format control string of the printf contains the conversion specification %s for printing a character string. So the preceding printf statement performs in essentially the same way as the preceding if…else statement.



```
printf( "%s\n", grade >= 60 ? "Passed" : "Failed" );
```

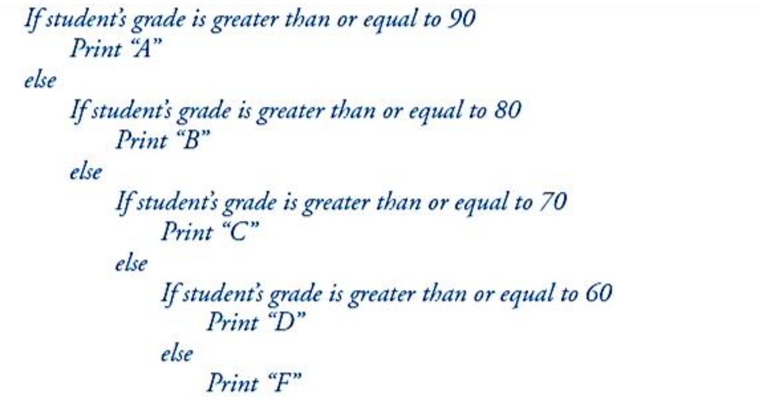
The second and third operands in a conditional expression can also be actions to be executed. For example, the conditional expression is read, "If grade is greater than or equal to 60 then printf("Passed\n"), otherwise printf( "Failed\n" )." This, too, is comparable to the preceding if…else statement. We'll see that conditional operators can be used in some situations where if…else statements cannot.



```
grade >= 60 ? printf( "Passed\n" ) : printf( "Failed\n" );
```

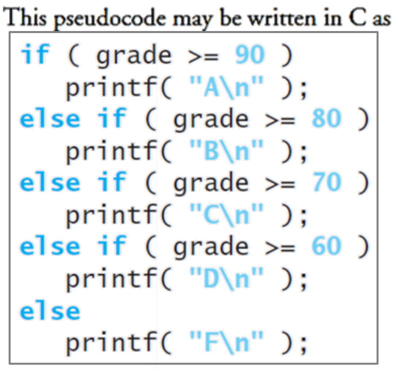
### Selection:  nested if

Nested if…else statements test for multiple cases by placing if..else statements inside if..else statements. For example, the following pseudocode statement will print A for exam grades greater than or equal to 90, B for grades greater than or equal to 80, C for grades greater than or equal to 70, D for grades greater than or equal to 60, and F for all other grades.



```
If student's grade is greater than or equal to 90
    Print "A"
else
    If student's grade is greater than or equal to 80
        Print "B"
    else
        If student's grade is greater than or equal to 70
            Print "C"
        else
            If student's grade is greater than or equal to 60
                Print "D"
            else
                Print "F"
```

This pseudocode may be written in C as



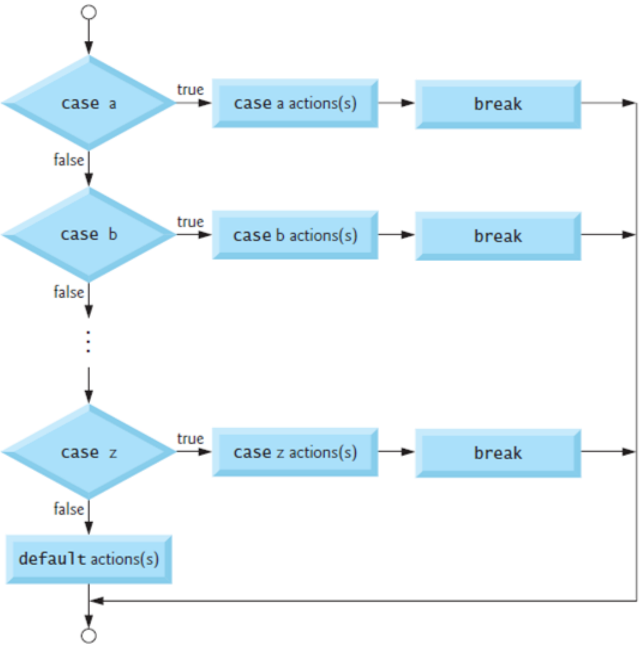
```c
if ( grade >= 90 )
   printf( "A\n" );
else if ( grade >= 80 )
   printf( "B\n" );
else if ( grade >= 70 )
   printf( "C\n" );
else if ( grade >= 60 )
   printf( "D\n" );
else
   printf( "F\n" );
```

### Selection:  Switch Case

Occasionally, an algorithm will contain a series of decisions in which a variable or expression is tested separately for each of the constant integral values it may assume, and different actions are taken. This is called multiple selection. C provides the switch multiple-selection statement to handle such decision making. The switch statement consists of a series of case labels, an optional default case and statements to execute for each case. Switch statement evaluate an expression by looking up for each case constant value. If an expression value matches with a case constant value then related statement/s is executed. If nothing match then default statement is executed. Expression and constant type should be integer (including char).

The example of switch case

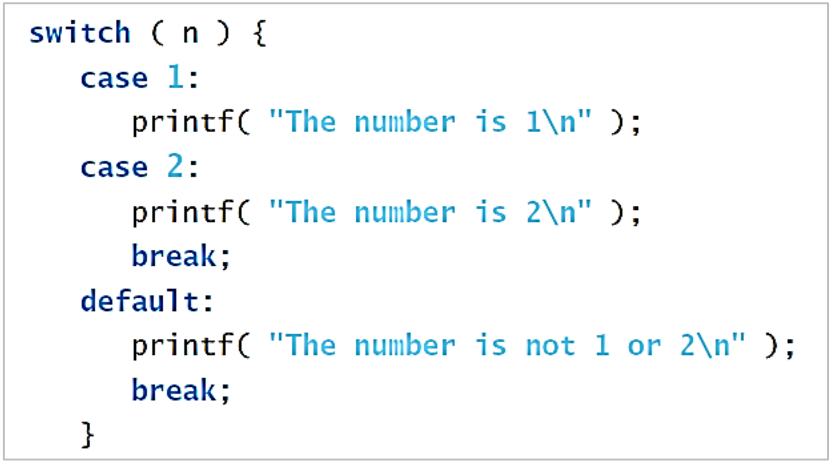
```c
#include <stdio.h>
int main()
{
    float value1, value2; char op;
    while(1) {
    printf(“\n Type value1 operator value2 \n”);
    scanf(“%f %c %f”, &value1, &op, &value2);
    switch(op){
       case(‘+’) : printf(“ = %f”, value1 + value2); break;
       case(‘-’) : printf(“ = %f”, value1 - value2); break;
       case(‘*’) : printf(“ = %f”, value1 * value2); break;
       case(‘/’) : printf(“ = %f”, value1 / value2); break;
       default :   printf(“ unknown operator!”);
    }
    }
    return(0);
}
```



## Soal
*Case*

- Create a C program to find the largest number among three numbers, using:
    - if
    - if...else
    - nested if

- Find the error in each of the following code segments and explain how to correct it.



```
switch ( n ) {
   case 1:
      printf( "The number is 1\n" );
   case 2:
      printf( "The number is 2\n" );
      break;
   default:
      printf( "The number is not 1 or 2\n" );
      break;
   }
```

- Solve all of this week’s exercises that are available on [https://socs1.binus.ac.id/quiz/](https://socs1.binus.ac.id/quiz/)

## References:

Paul Deitel & Harvey Deitel. (2016). C how to program:with an introduction to C++. 08. Pearson Education. Hoboken.ISBN: 9780133976892.
Choosing between Alternatives: http://docs.roxen.com/pike/7.0/tutorial/statements/conditions.xml
Getting Controls: http://aelinik.free.fr/c/ch10.htm