C Programming

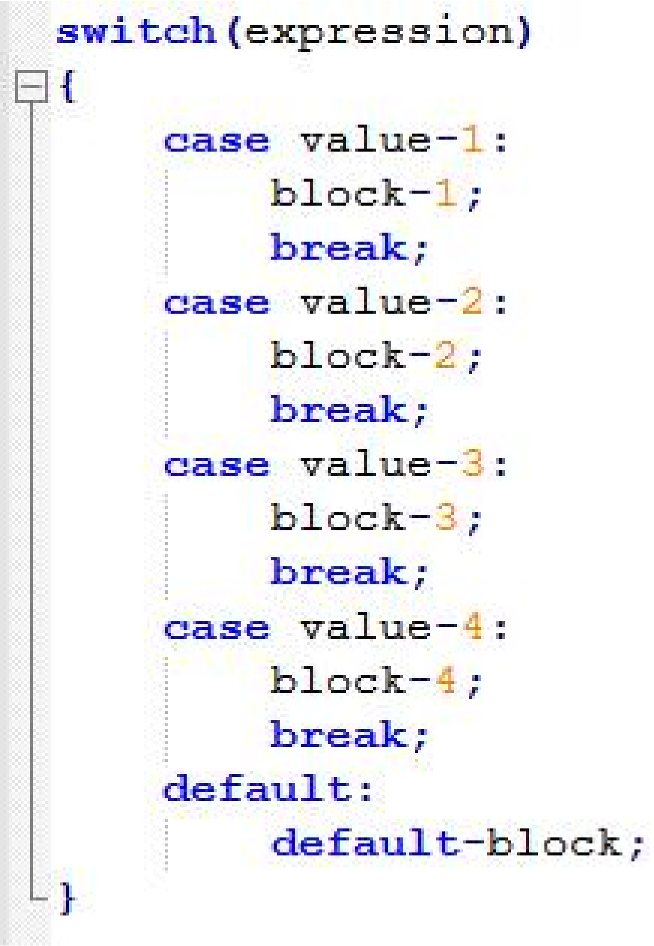
Switch Statement

Dr. Asif Uddin Khan

Switch statement in C

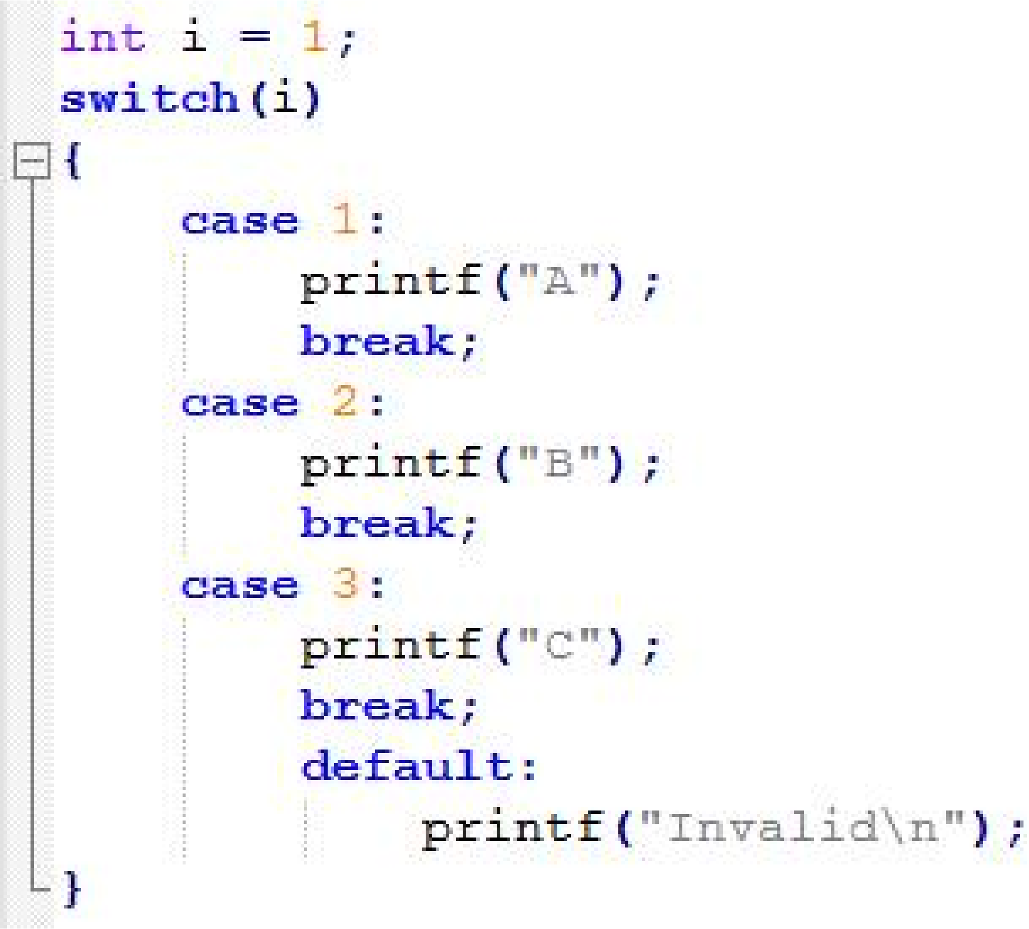
- Allows us to choose only one choice among the many given choices.
- Used in Menu like program, where one value is associated with each option and you need to choose only one at a time.
- The expression in switch evaluates to return an integral value, which is then compared to the values present in different cases.
- It executes that block of code which matches the case value.
- If there is no match, then **default** block is executed(if present).

Syntax



```
switch(expression)
{
        case value-1:
            block-1;
            break;
        case value-2:
            block-2;
            break;
        case value-3:
            block-3;
            break;
        case value-4:
            block-4;
            break;
        default:
            default-block;
}
```

Program-1:Switch statement in C

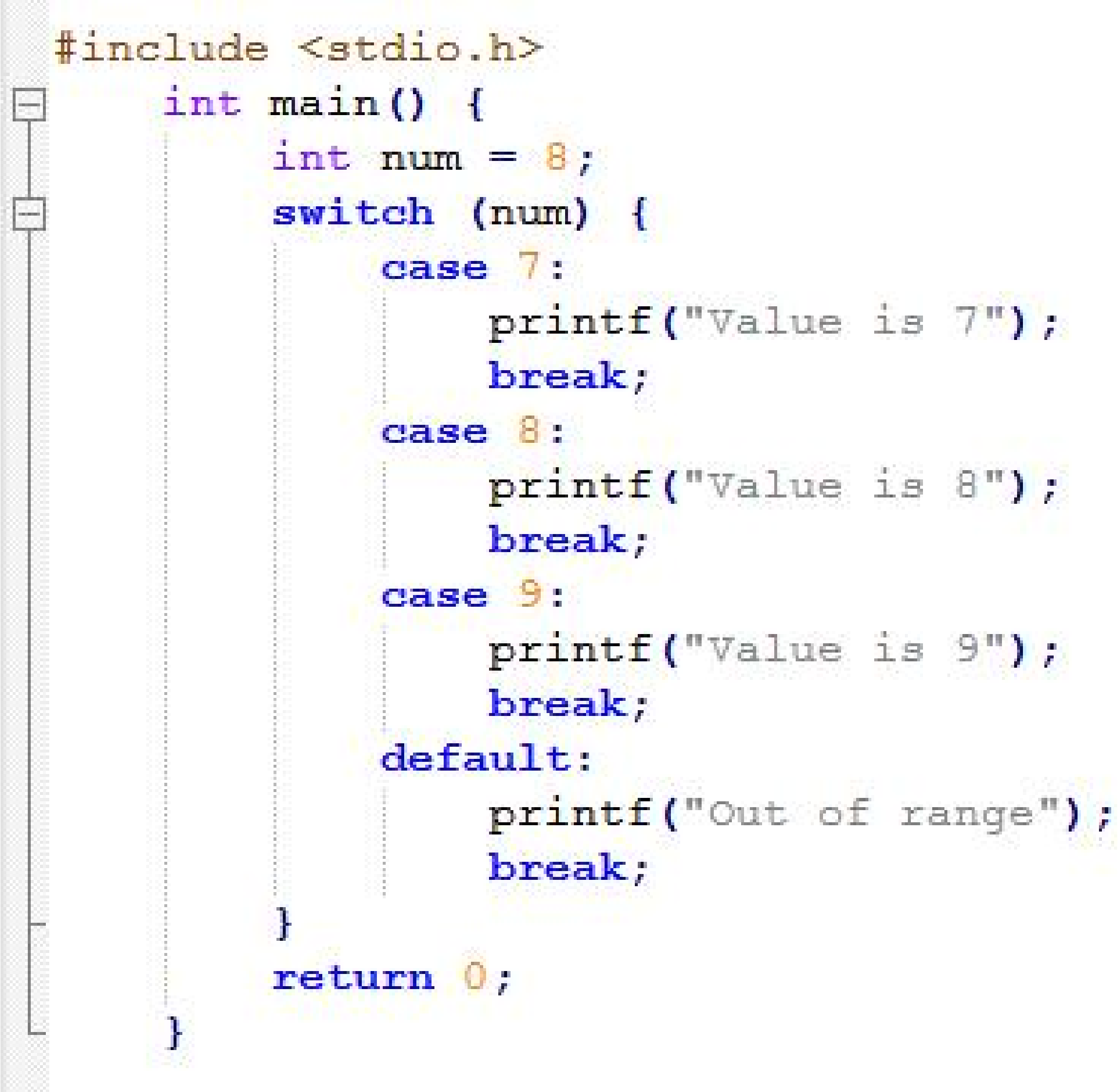


```
int i = 1;
switch(i)
{
    case 1:
        printf("A");
        break;
    case 2:
        printf("B");
        break;
    case 3:
        printf("C");
        break;
        default:
            printf("Invalid\n");
}
```

Output

A

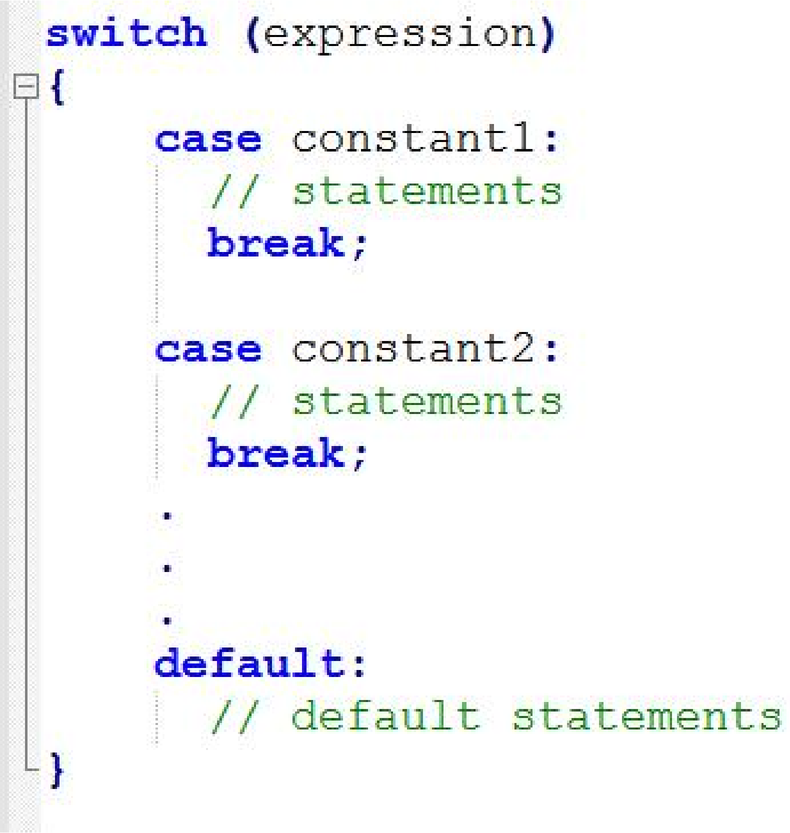
Program-2 using Switch case statement



```
#include <stdio.h>
    int main() {
        int num = 8;
        switch (num) {
            case 7:
                printf("Value is 7");
                break;
            case 8:
                printf("Value is 8");
                break;
            case 9:
                printf("Value is 9");
                break;
            default:
                printf("Out of range");
                break;
        }
        return 0;
    }
```

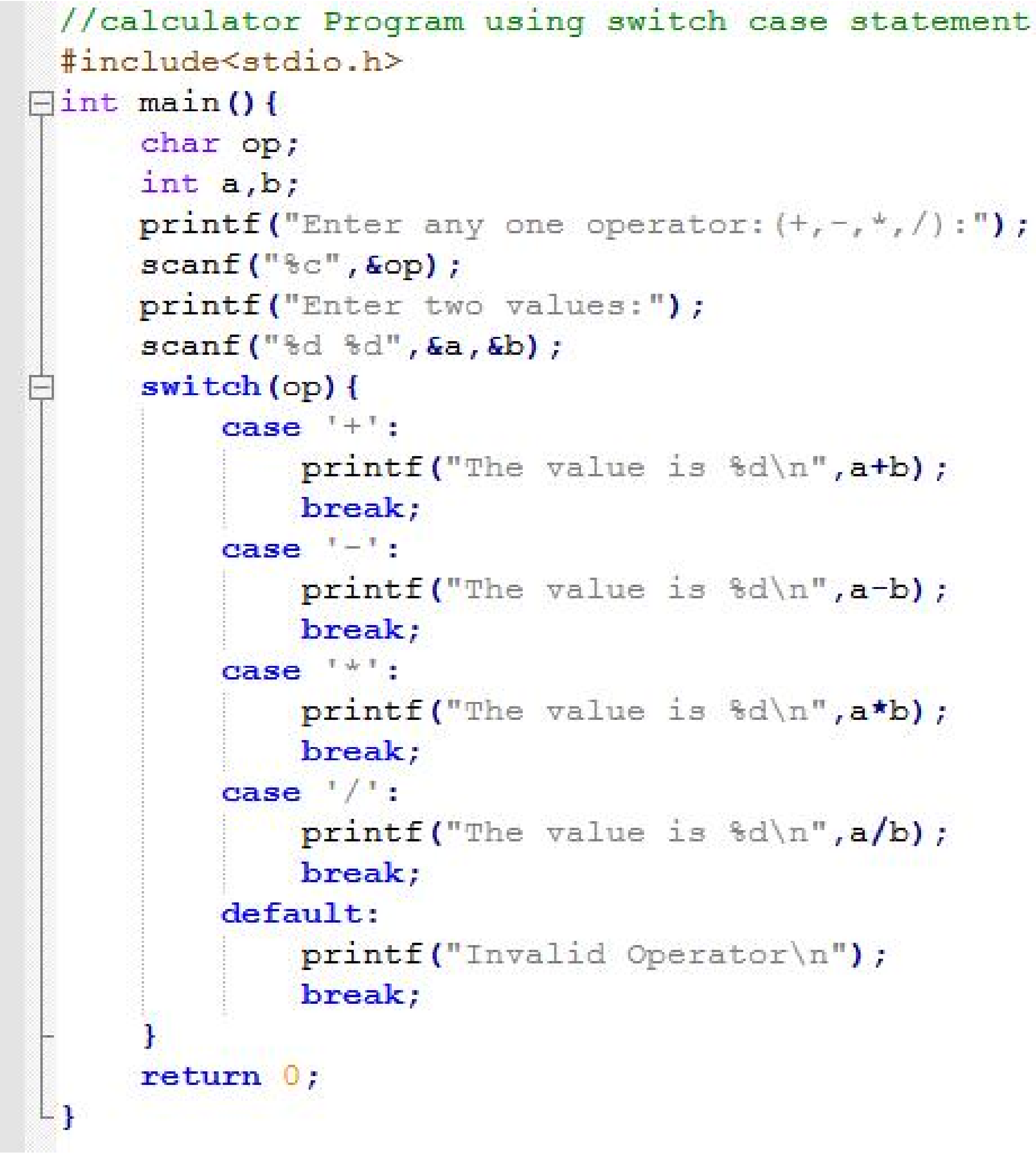
How does the switch statement work?

- The expression is evaluated then statements after the matching case are executed.
- For example, if the value of the expression is equal to constant2, statements after case constant2: are executed until break is encountered.
- If there is no match, the default statements are executed.
- If we do not use break, all statements after the matching label are executed.
- By the way, the default clause inside the switch statement is optional.



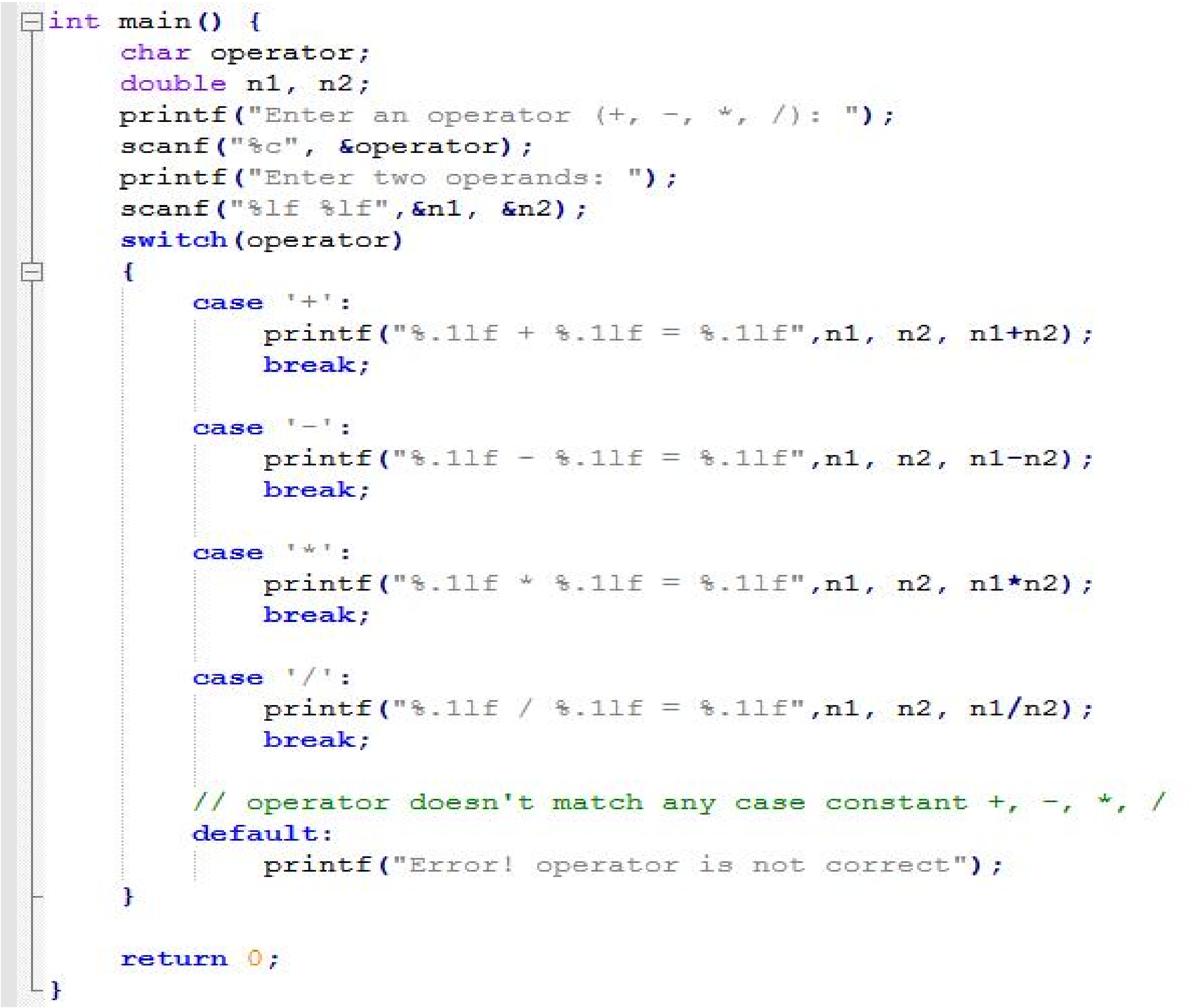
```
switch (expression)
{
      case constant1:
        // statements
        break;

      case constant2:
        // statements
        break;
      .
      .
      .
      default:
        // default statements
}
```

Calculator program-1 using switch case



```c
//calculator Program using switch case statement
#include<stdio.h>
int main(){
    char op;
    int a,b;
    printf("Enter any one operator:(+,-,*,/):");
    scanf("%c",&op);
    printf("Enter two values:");
    scanf("%d %d",&a,&b);
    switch(op){
        case '+':
            printf("The value is %d\n",a+b);
            break;
        case '-':
            printf("The value is %d\n",a-b);
            break;
        case '*':
            printf("The value is %d\n",a*b);
            break;
        case '/':
            printf("The value is %d\n",a/b);
            break;
        default:
            printf("Invalid Operator\n");
            break;
    }
    return 0;
}
```

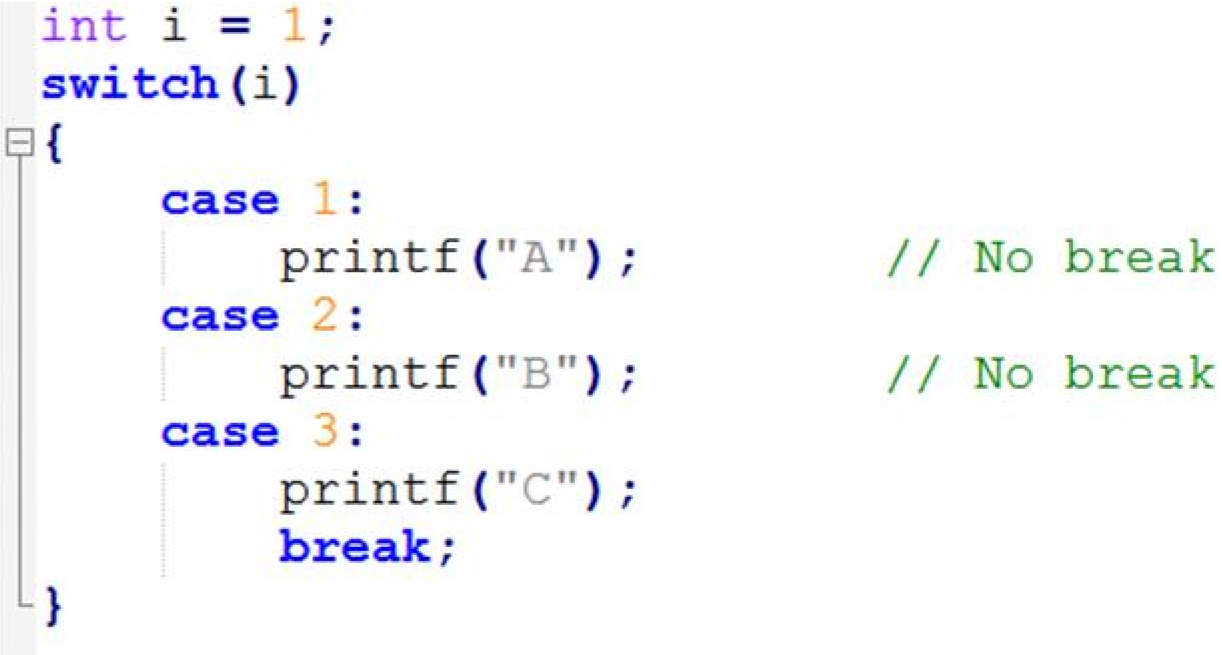
Calculator program-2 using switch case



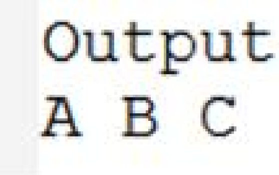
```
int main() {
    char operator;
    double n1, n2;
    printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &operator);
    printf("Enter two operands: ");
    scanf("%lf %lf",&n1, &n2);
    switch(operator)
    {
        case '+':
            printf("%.1lf + %.1lf = %.1lf",n1, n2, n1+n2);
            break;

        case '-':
            printf("%.1lf - %.1lf = %.1lf",n1, n2, n1-n2);
            break;

        case '*':
            printf("%.1lf * %.1lf = %.1lf",n1, n2, n1*n2);
            break;

        case '/':
            printf("%.1lf / %.1lf = %.1lf",n1, n2, n1/n2);
            break;

        // operator doesn't match any case constant +, -, *, /
        default:
            printf("Error! operator is not correct");
    }

    return 0;
}
```

Example

If we do not use break, all statements after the matching label are executed.



```
int i = 1;
switch(i)
{
    case 1:
        printf("A");            // No break
    case 2:
        printf("B");            // No break
    case 3:
        printf("C");
        break;
}
```



```
Output
A B C
```

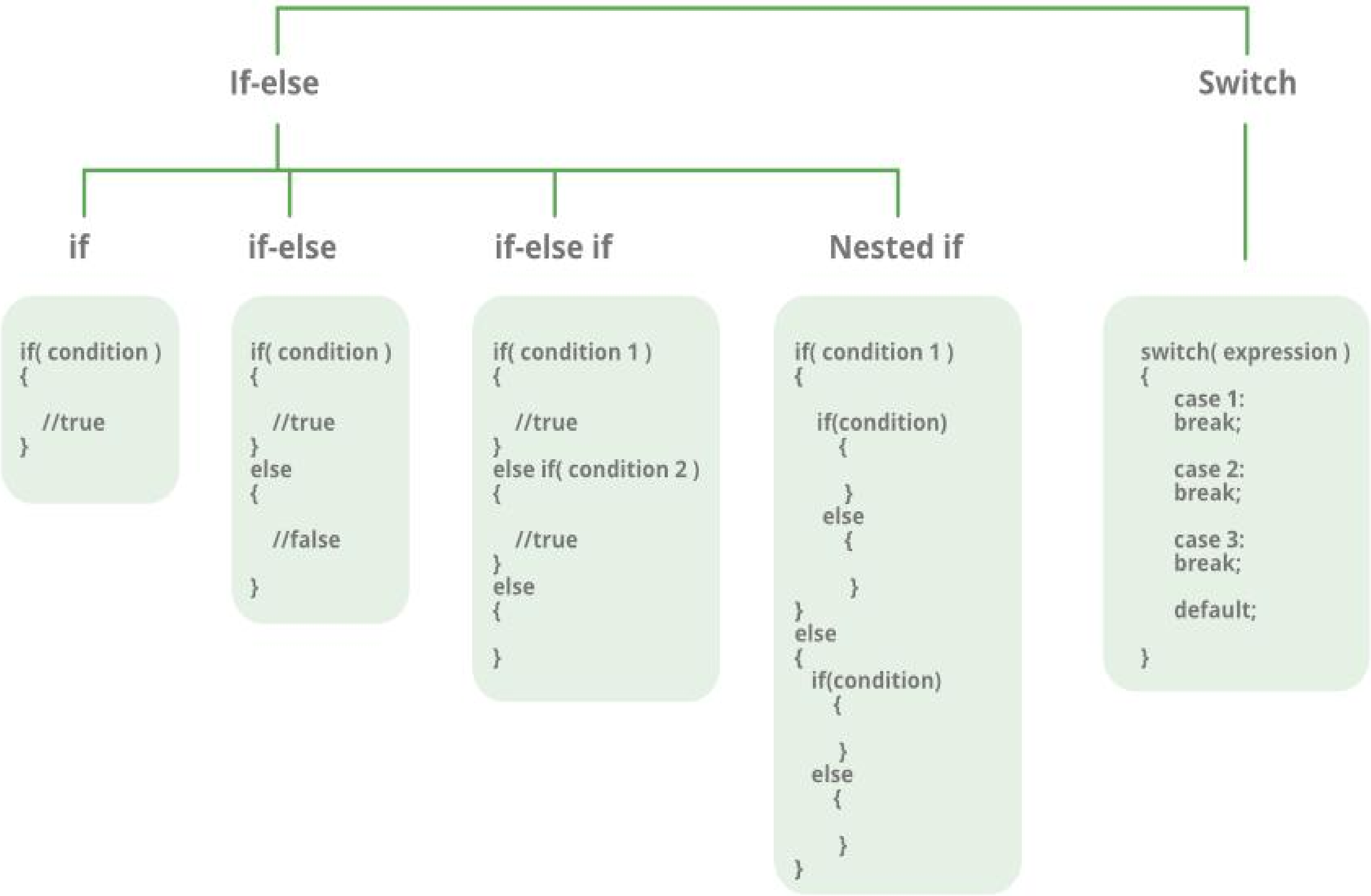
Rules for using switch statement

- The expression (after switch keyword) must yield an **integer** value i.e the expression should be an integer or a variable or an expression that evaluates to an integer.
- The case **label** values must be unique.
- The case label must end with a colon(:)
- The next line, after the **case** statement, can be any valid C statement.

Points to Remember

- break statements are used to **exit** the switch block.
- It isn't necessary to use break after each block, but if you do not use it, then all the consecutive blocks of code will get executed after the matching block.

Decision Making



If-else

Switch

if

```
if( condition )
{

   //true
}
```

if-else

```
if( condition )
{

   //true
}
else
{

   //false

}
```

if-else if

```
if( condition 1 )
{

   //true
}
else if( condition 2 )
{

   //true
}
else
{

}
```

Nested if

```
if( condition 1 )
{
  if(condition)
     {

     }
   else
     {

     }
}
else
{
  if(condition)
     {

     }
  else
     {

     }
}
```

```
switch( expression )
{
     case 1:
     break;

     case 2:
     break;

     case 3:
     break;

     default;

}
```