# Introduction to Programming
## Lecture Switch Statement

**Muhammad Salman**

ABASYN UNIVERSITY

## Summary

- Switch Statement
- First Program
- Break Statement
- Program
- Default statement
- goto Statment

## Switch statment

- It is the statement that allows a variable to be tested against a list of values for equality. The value in the switch is termed as a case, and hence the variable being switched on is checked against the case.

- Switch statement is used as a substitute for a “nested if-else” statement. It is used when multiple choice is given and one choice is to be made.

- Nested if-else structure becomes complicated in multiple choice. The switch statement is used in such cases.

- The expression in the switch statement of C++ is valid only if it results in a constant value.
- The switch statement evaluates an expression and returns a value.
- The returned value is compared with the constant values given in the case.

# The switch statement flow digram

## Syntax

```
switch (Expression)
{
case 1 :
    statement 1 ;
case 2 :
    statement  2 ;
case N :
    statement n ;
default:
    statement d ;

}
```

# C++ Switch Statement Program

```cpp
#include <iostream>
using namespace std;
int main()
{
char grade;
cout <<“Enter a grade”<<endl;
cin >>grade;
switch (grade)
{
    case 'A' :
        cout <<"Excellent" <<endl;
    case 'B' :
        cout << "Very Good"<<endl ;
    case 'C' :
        cout << "Good"<<endl ;
    case 'D' :
        cout << "Poor"<<endl ;
    case 'F' :
        cout << "Fail"<<endl ;
}
}
```

```
Enter a grade
A
Excellent
Very Good
Good
Poor
Fail
```

- It is the statement that allows a variable to be tested against a list of values for equality (A,B,C,D,F). The value in the switch is termed as a case, and hence the variable being switched on is checked against the case.

# C++ Switch Statement Program to match the case

```cpp
#include <iostream>
using namespace std;

int main()
{
int x;
cin>>x;
switch (x) {
    case 0:
        cout <<"Enter value x"<<endl;
        cout<<"You have entered 0"<<endl; // Exact match
        break;
    case 1: // Exact match
        cout <<"Enter value x"<<endl;
        cout<<"You have entered 1"<<endl;
        break;
    case 2:
        cout <<"Enter value x"<<endl;
        cout<<"You have entered 2 "<<endl;
}
}
```

**Input:**

**cin >> x; reads an integer value from the user and stores it in x.**

**Switch Statement:**

**The switch statement evaluates the value of x.**

**If x is 0, the case 0: block is executed, and it prints "You entered 0".**

**If x is 1, the case 1: block is executed, and it prints "You entered 1".**

**If x is 2, the case 2: block is executed, and it prints "You entered 2".**

## Break statement

- The break statement interrupts the flow of control.
- We have seen in the last example that even the true case was found but the flow of control went through all the statements.
- We want only the true case should be executed.
- For this purpose, we use the **“break statement”**
- Write it after the case, thus when a true case is found execute the statement then the break statement interrupts the flow of control, and control jumps out of the switch statement.
- If it is not used, then the statement of other cases that come after the matching case will also execute.

# Break statement

```
#include <iostream>
using namespace std;
int main()
{
char grade;
cout <<“Enter a grade”<<endl;
cin >>grade;
switch (grade)
{
    case 'A' :
        cout <<"Excellent" <<endl;
    case 'B' :
        cout << "Very Good"<<endl ;
    case 'C' :
        cout << "Good"<<endl ;
    case 'D' :
        cout << "Poor"<<endl ;
    case 'F' :
        cout << "Fail"<<endl ;
}
}
```

```
Enter a grade
a


=== Code Execution Successful ===
```

**Yes !!!‘A’ is different from ‘a’**

**case ‘A’ :**
**case ‘a’ :**
**cout <<“Excellent” << endl;**

**Refer to the next slide**

# Break statement

```cpp
using namespace std;
int main()
{
char grade;
cout <<“Enter a grade”<<endl;
cin >>grade;
switch (grade)
{
    case 'A' :
    case 'a' :
        cout <<"Excellent" <<endl;
        break;
    case 'B' :
    case 'b' :
        cout << "Very Good"<<endl ;
        break;
    case 'C' :
    case 'c' :
        cout << "Good"<<endl ;
        break;
    case 'D' :
    case 'd' :
        cout << "Poor"<<endl ;
        break;
    case 'F' :
    case 'f' :
        cout << "Fail"<<endl ;
}
}
```

```
Enter a grade
f
Fail
```

# C++ Switch Statement Program to  find an even number

```cpp
// The case 0 is specifically designed to handle the situation where n is e
//When n % 2 evaluates to 0, the code inside the case 0 block will execute.


#include <iostream>
using namespace std;
int main()
{
int n;
cout <<“Enter a Number”<<endl;
cin >>n;
switch (n%2)
{
case 0 :
    cout <<"Divisiable by 2"<<endl;
    break;
case 1 :
    cout <<"Not Divisiable by 2 "<<endl;
    break;
}
cout <<"Out of Switch statment"<<endl;
return 0;
}
```

**Output**

```
Enter a Number
10
Divisiable by 2
Out of Switch statment
```

**Output**

```
Enter a Number
11
Not Divisiable by 2
Out of Switch statment
```

## n % 2 Expression:

- The expression n % 2 calculates the remainder when n is divided by 2.
- If n is even, n % 2 will be 0.
- If n is odd, n % 2 will be 1.

switch Statement:

The switch statement evaluates the result of n % 2.

case 0: corresponds to even numbers.

case 1: corresponds to odd numbers.

~~case 2:~~

~~case 3:~~

- The case labels must match the possible results of n % 2, which are 0 and 1.

case 0:

- If n % 2 is 0, the program executes the block under case 0 and prints that n is even.

case 1:

- If n % 2 is 1, the program executes the block under case 1 and prints that n is odd.

## Default case

- The default case in a switch statement acts similarly to an else statement in an if-else block.
- It is executed when none of the other cases match the value of the expression being evaluated.

- Comparison:
- The else block is executed when none of the proceedings if or else if conditions is true

# Default case

```cpp

#include <iostream>
using namespace std;

int main() {
    int day;

    cout << "Enter a number (1-3) to represent the day of the week: ";
    cin >> day;

    switch (day) {
        case 1:
            cout << "Monday" << endl;
            break;
        case 2:
            cout << "Tuesday" << endl;
            break;
        case 3:
            cout << "Wednesday" << endl;
            break;
        default:
            cout << "Invalid input! Please enter a number between 1 and 7." <<
                endl;
            break;
    }
```

**Output**

```
Enter a number (1-3) to represent the day of the week: 9
Invalid input! Please enter a number between 1 and 7.


=== Code Execution Successful ===
```

# Basic Arithmetic operation using switch statement

```cpp
#include <iostream>
using namespace std;
int main() {
    char operation;
    int num1, num2;

    cout << "Enter first Num , operator (+, -, *, /): , 2nd Number"<<endl;

    cin >> num1 >> operation>> num2;

    switch (operation) {
        case '+':
            cout << num1 << " + " << num2 << " = " << num1 + num2 << endl;
            break;
        case '-':
            cout << num1 << " - " << num2 << " = " << num1 - num2 << endl;
            break;
        case '*':
            cout << num1 << " * " << num2 << " = " << num1 * num2 << endl;
            break;
        default:
            cout << "Invalid operator! Please enter +, -, *, or /." << endl;
            break;
    }
    return 0;
```

**Output**

```
Enter first Num , operator (+, -, *, /): , 2nd Number
50
+
50
50 + 50 = 100
```

# Basic Arithmetic operation using switch statement

```cpp
#include <iostream>
using namespace std;
int main() {
    char operation;
    int num1, num2;

    cout << "Enter first Num , operator (+, -, *, /): , 2nd Number"<<endl;

    cin >> num1 >> operation>> num2;

    switch (operation) {
        case '+':
            cout << num1 << " + " << num2 << " = " << (num1 + num2) << endl;
            break;
        case '-':
            cout << num1 << " - " << num2 << " = " << (num1 - num2) << endl;
            break;
        case '*':
            cout << num1 << " * " << num2 << " = " << (num1 * num2) << endl;
            break;
        default:
            cout << "Invalid operator! Please enter +, -, *, or /." << endl;
            break;
    }
    return 0;
```

```
Enter first Num , operator (+, -, *, /): , 2nd Number
5
-
10
5 - 10 = -5
```

# Salary calculation using switch statement

```
#include <iostream>
using namespace std;
int main ( )
  {
int salary ;
float deduction, netPayable ;
cout << "Please enter the salary"<<endl; ;
cin >> salary ;
switch (salary/60000)
 {
case 0 :
cout <<salary/60000<<endl;

deduction = 0; // as deduction is zero in this case
netPayable = salary ;

 cout <<"NetPayable (salary - deduction) ="<< endl;
 cout <<"Net Payable (salary - deduction)=" <<endl;
 cout << salary << " - "<< deduction << " = " <<netPayable<<endl;
  break;
case 1 :
cout <<salary/60000<<endl;
deduction =  salary*5/100 ;
netPayable = salary - deduction ;
// this means salary is in range 60000 - 120000
 cout << "Net Payable (salary - deduction)  = " <<endl ;
 cout << salary << " - " << deduction << " = " << netPayable;
  break;
 //necessary to exit switch
default :
// this means the salary is 20,000 or more
cout <<salary/60000<<endl;
deduction =  salary *  15 /100 ;
netPayable = salary-deduction ;
//cout << "Net Payable (salary - deduction)  =  " ;
cout << salary <<  " - " << deduction << " = " << netPayable;
}
    return 0;
}
```

- If salary is less than 60000 no tax will deduct.
- If salary slab is 60000 -119000 then 5 % tax will deduct ,
- if salary slab is 120000 and above 15 % tax will deduct

## goto statment

- The goto is an unconditional branch of execution.
- goto statement is used to jump the control anywhere (back and forth ) in the program.

# goto statment

```cpp
#include <iostream>
using namespace std;

int main() {
    int day;

    cout << "This Progrm will Demonstrate the goto statment"<<endl;
    cout <<"Testing"<<endl;
    goto abc;
    cout <<"it will print later"<<endl;
    cout << "This another c++ statment"<<endl;
    abc:
    cout <<"Program ended ???"<<endl;
    cout <<"Yes";

    return 0;
}
```

```
This Progrm will Demonstrate the goto statment
Testing
Program ended ???
Yes
```

# Thanks