Al-Mustaqbal University

College of Engineering & Technology

Biomedical Engineering Department

Computer

Lecture 7 & 8

Switch, Break and continue statements

Dr. Ahmed Hasan Janabi
PhD in Cybersecurity
Email: Ahmed.Janabi@uomus.edu.iq

break Statement

- In C programming, break is used in terminating the loop immediately after it is encountered. The break statement is used with conditional if statement.

- Syntax of break statement

```
break;
```

Flow Chart Of Break Statement

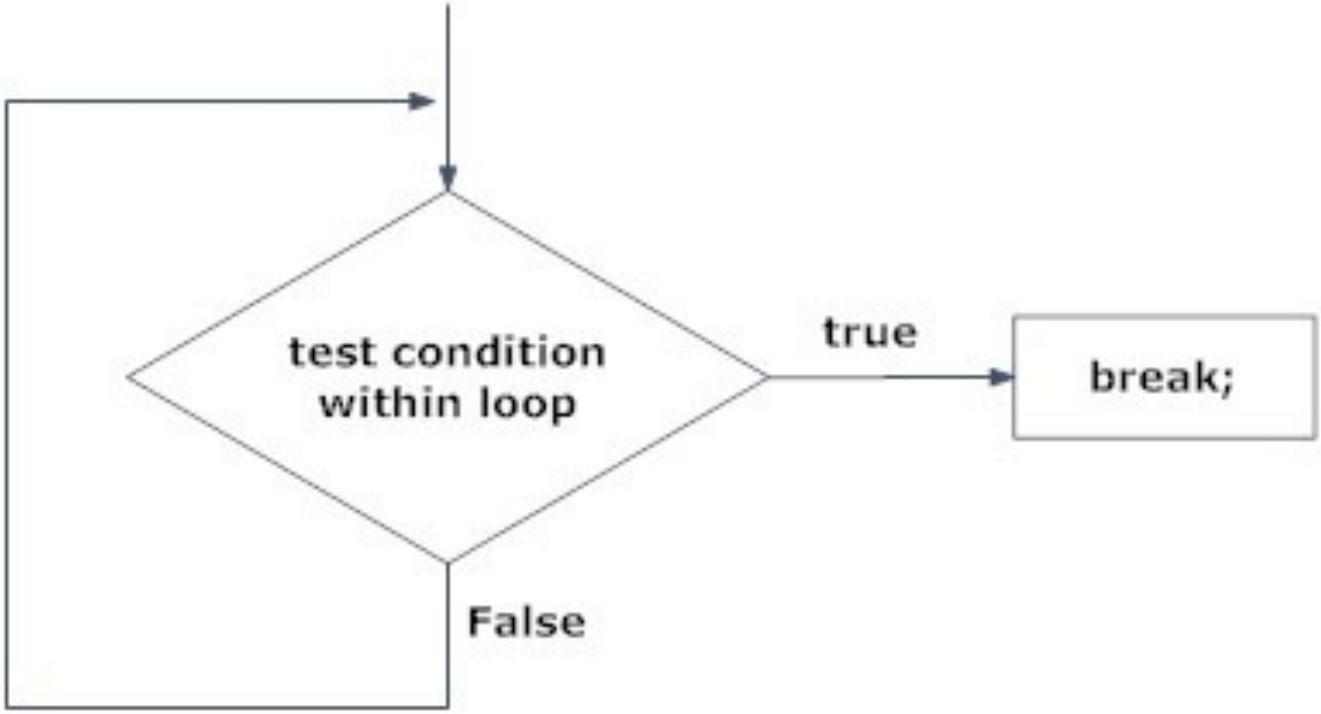

Figure: Flowchart of break statement

Example of break statement

Write a C program to find average of maximum of *n* positive numbers entered by user. But, if the input is negative, display the average(excluding the average of negative input) and end the program.

```
# include <stdio.h>
int main()
{
 float num, average, sum;
int i,n;
printf("Maximum no. of inputs\n");
scanf("%d",&n);
for(i=1;i<=n;++i)
{
printf("Enter n%d: ",i);
scanf("%f",&num);
if(num<0.0)
break;
 sum=sum+num;
}
average=sum/(i-1);
printf("Average=%.2f",average);
return 0;
}
```

Output

Maximum no. of inputs 4 Enter n1: 1.5 Enter n2: 12.5 Enter n3: 7.2 Enter n4: -1 Average=7.07

continue Statement

- It is sometimes desirable to skip some statements inside the loop. In such cases, continue statements are used.
- Syntax of continue statement

```
continue;
```

Flow Chart Of Continue Statement

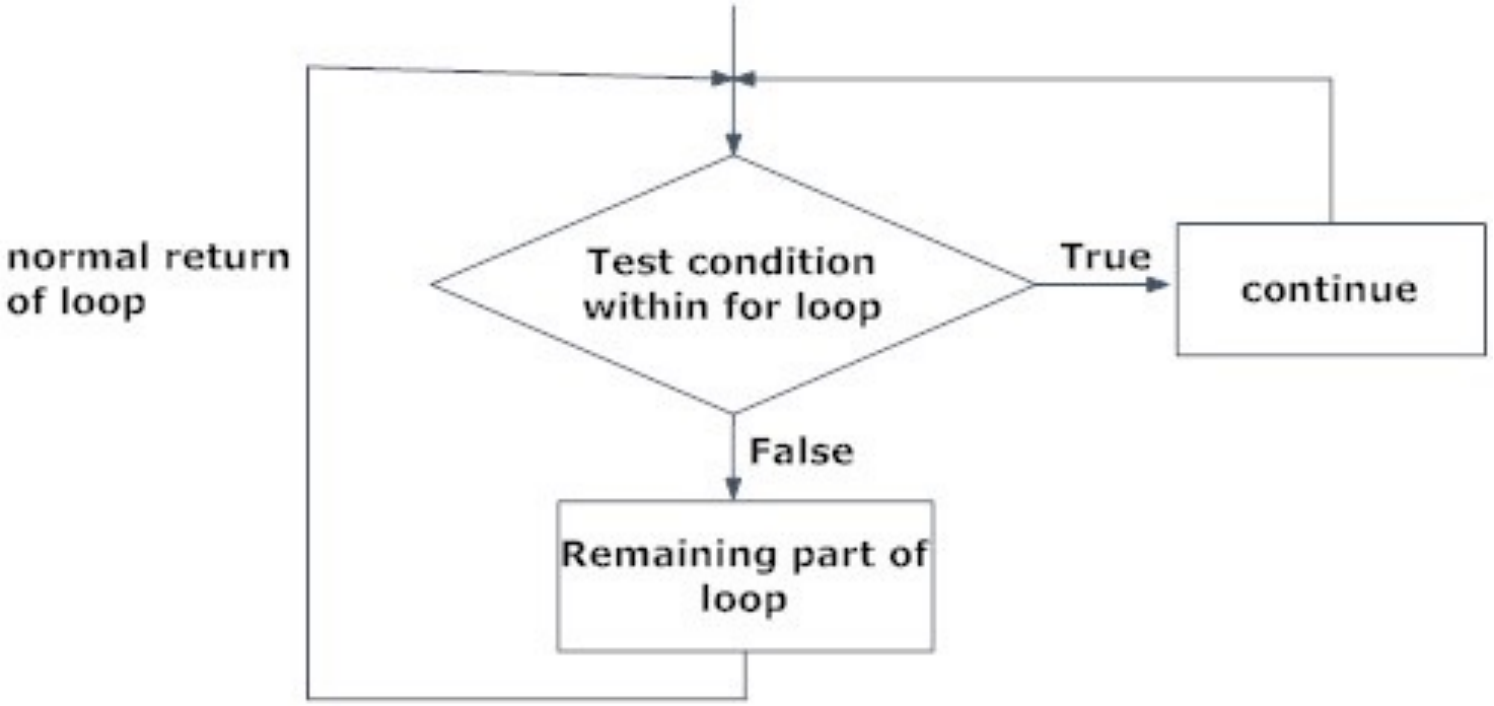

Fig: Flowchart of continue statement

Example of continue statement

Write a C program to find the product of 4 integers entered by a user. If user enters 0 skip it.

```
 # include <stdio.h>
 int main()
{
 int i,num,product;
 for(i=1,product=1;i<=4;++i)
{
 printf("Enter num%d:",i);
 scanf("%d",&num);
 if(num==0)
 continue;
 product*=num;
 }
 printf("product=%d",product);
}
```

➢**Output:**

Enter num1:3 Enter num2:0 Enter num3:-5 Enter num4:2 product=-30

C Programming switch Statement

- Decision making are needed when, the program encounters the situation to choose a particular statement among many statements. If a programmer has to choose one block of statement among many alternatives, nested if…else can be used but, this makes programming logic complex. This type of problem can be handled in C programming using switch statement.

Syntax of switch...case

switch (*n*) {
case constant1:
code/s to be executed if *n* equals to *constant1*; break;
case *constant2*:
code/s to be executed if *n* equals to *constant2*; break; . . .
default:
code/s to be executed if *n* doesn't match to any cases;
}

Flow Chart Of Switch...Case

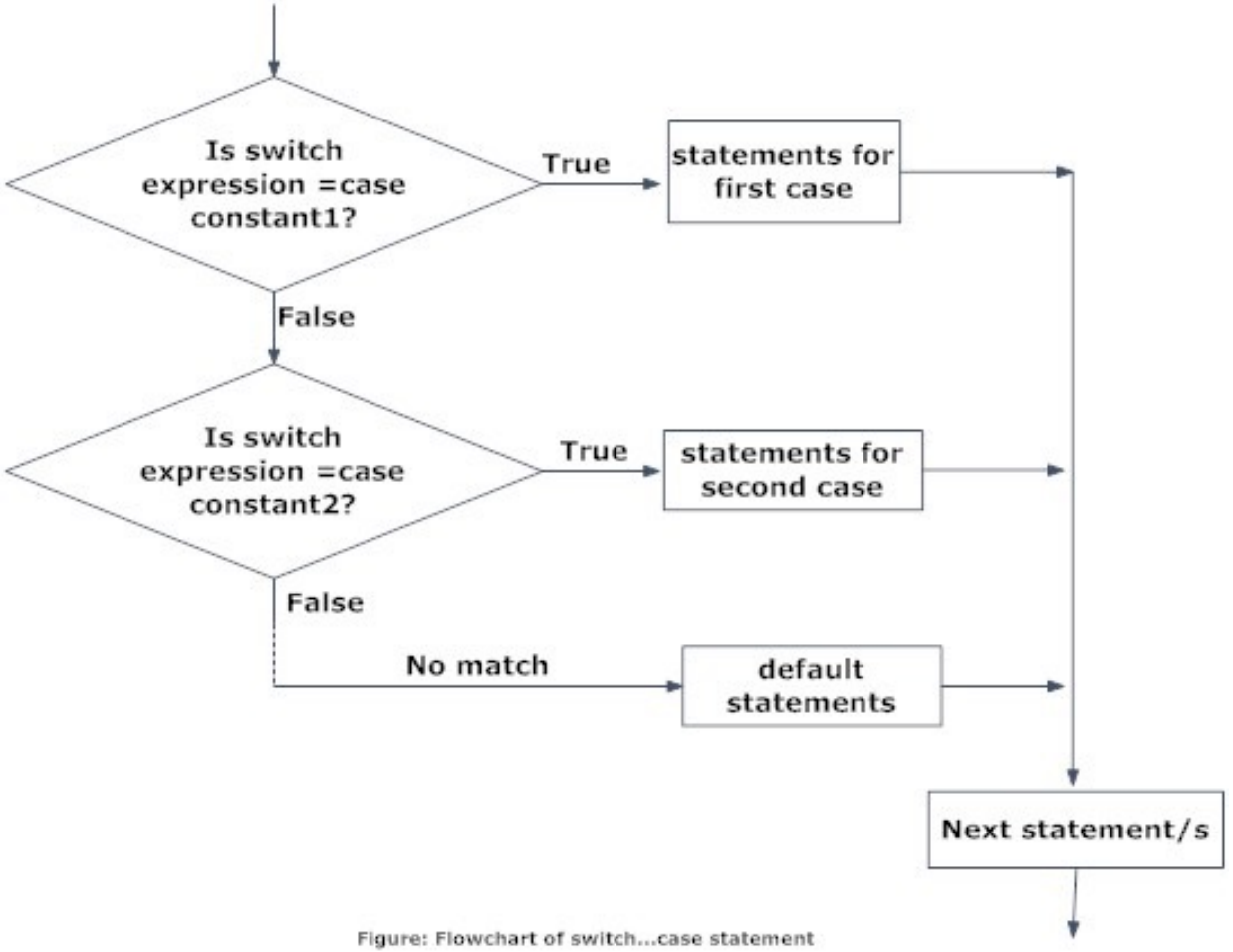

Figure: Flowchart of switch...case statement

Example of switch...case statement

Write a program that asks user to select an arithmetic operator('+','-','*' or '/') and two operands and perform the corresponding calculation on the operands.

```
# include <stdio.h>
 int main()
{
char o;
float num1,num2;
printf("Select an operator either + or - or * or / \n");
 scanf("%c",&o);
 printf("Enter two operands: ");
 scanf("%f%f",&num1,&num2);
switch(o) {
 case '+':
printf("%.1f + %.1f = %.1f",num1, num2, num1+num2);
 break;
case '-':
```

(CONT.)

```
printf("%.1f - %.1f = %.1f",num1, num2, num1-num2);
break;
case '*':
printf("%.1f * %.1f = %.1f",num1, num2, num1*num2);
break;
case '/':
printf("%.1f / %.1f = %.1f",num1, num2, num1/num2);
break;
default:
printf("Error! operator is not correct");
break; }
}
```

➢**Output:**

Enter operator either + or - or * or / * Enter two operands: 2.3 4.5 2.3 * 4.5 = 10.3

Exercises

Exercise 1: Write a C++ program that uses a loop to find the first multiple of 5 between 1 and 50. Use the break statement to exit the loop once the multiple is found.

```
#include <iostream>
using namespace std;
int main() {
   for (int i = 1; i <= 50; ++i) {
      if (i % 5 == 0) {
         cout << "The first multiple of 5 between 1 and 50 is: " << i << endl;
         break;
      }
   }
}
```

Exercises

Exercise 2: Write a C++ program that uses a loop to print all numbers from 1 to 10, except the number 4. Use the continue statement to skip the number 4.

```
#include <iostream>
using namespace std;
int main() {
   for (int i = 1; i <= 10; ++i) {
      if (i == 4) {
         continue;
      }
      cout << i << " ";
   }
   cout << endl;
}
```

Exercises

Exercise 3: Write a C++ program that reads an integer from the keyboard and prints the corresponding day of the week. Use a switch statement to handle the different cases (1 for Monday, 2 for Tuesday, etc.).

```
#include <iostream>
using namespace std;
int main() {
   int day;
   cout << "Enter a number (1-7) to get the corresponding 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;
```

Exercises

Exercise 3: (Cont.)

```
 case 4:
        cout << "Thursday" << endl;
        break;
    case 5:
        cout << "Friday" << endl;
        break;
    case 6:
        cout << "Saturday" << endl;
        break;
    case 7:
        cout << "Sunday" << endl;
        break;
    default:
        cout << "Invalid input! Please enter a number between 1 and 7." << endl;
  }
}
```

Exercises

Exercise 4: Write a C++ program that reads a student's score (0-100) from the keyboard and determines their grade based on the score. Use a switch statement to handle the different grade ranges (A, B, C, D, F).

```
#include <iostream>
using namespace std;
int main() {
   int score;
   char grade;
   cout << "Enter the student's score (0-100): ";
   cin >> score;

   switch (score / 10) {
      case 10:
      case 9:
         grade = 'A';
         break;
      case 8:
         grade = 'B';
         break;
```

Exercises

Exercise 4: (Cont.)

```
 case 7:
          grade = 'C';
          break;
       case 6:
          grade = 'D';
          break;
       default:
          grade = 'F';
          break;
    }
    cout << "The grade is: " << grade << endl;

}
```

HomeWorks:

- **Task**: Write a C++ program that reads student scores from the keyboard until a negative score is entered. Use a switch statement to determine the grade based on the score and display the result. Use continue to skip invalid scores (scores not in the range 0-100).
- **Requirements**:

1. Continuously read scores from the user until a negative score is entered.
2. Use continue to skip invalid scores (less than 0 or greater than 100).
3. Use a switch statement to assign a grade (A, B, C, D, F) based on the score range.
4. Use break to exit the loop when a negative score is entered.

THANK YOU