Certificate - Software Development

Control Constructs in C Language

If/else Statement

As we learnt earlier if/else statement is used to make selections. Lets' see the syntax of this in C language.

Basic syntax of if / else

```
 if (expression)
     statement_1;
[else
     statement_2;]
```

Example 1

```
if (ch == ‘A’)
    printf(“It’s Letter A\n”);
```

Example 2

```
if ( num % 2 == 1 )
    printf(“Num is Odd\n”);
else
    printf(“%d is Even\n”, num);
```

If/else Syntax - with Statement Block

When if or else section has more than 1 statement is has to be marked as a block of statement using { }

Syntax of if/else

```
if (expression)
{
    Statement_1;
    Statement_2;
}
else {
    Statement_3;
    Statement_4;
}
```

Example 3

```
 if ( n>0 )
     avg = sum / n;
 else {
     printf(“Can’t compute average\n”);
     avg = 0;
}
```

Nested If and Switch Statement

Nested IF Example

```
main()
{ int marks, grade;
  printf("Enter your marks : ");
  scanf("%d",&marks)
  if (marks >=80)
     grade = 'D';
  else if (marks >= 60)
     grade = 'C';
  else if (marks >= 40)
     grade = 'P';
  else
     grade = 'F';
  printf("Grade = %c\n", grade);
}
```

Switch Example

```
main()
{
  int num;
  printf("Enter an integer : ");
  scanf("%d",&num);
  switch (num)
  {
     case 1: printf("One\n"); break;
     case 2: printf("Two\n"); break;
     case 3: case 4:
          printf("Three or Four\n"); break;
     default : printf("Unknown\n");
  }
}
```

While Loop

The most basic loop in C is the while loop. A while loop has one control expression, and executes as long as that expression is 1 (true).

Syntax of While Loop

```
while  (expression)
{
     Statement_1;
     Statement_2;
     Statement_3;
}
```

While Loop Example

```
main()
{
    int num=1, sum=0;
    while ( num<=10 )
    {
        sum = sum + num;
        num = num + 1;
    }
    printf("Sum = %d\n",sum);
}
```

For Loop

For loop uses a syntax where initialization, condition and the change all are given in one embedded block.

Syntax of For Loop

```
for (initialize; condition; change)
{
     Statement_1;
     Statement_2;
     Statement_3;
}
```

While Loop Example

```
main()
{
   int num, sum=0;
   for (num=1; num<=10; num = num +1)
   {
      sum = sum + num;
      //num = num + 1; this is not needed
   }
   printf("Sum = %d\n",sum);
}
```

Do While Loops

This is a small restructured version of while loop which allows the code to be executed at least once even if the condition is initially false.

- **While Loop**
  - Condition at the beginning
  - Test and process
  - Block can be skipped
- **Do While Loop**
  - Condition at the bottom
  - Process and test
  - Block is done at least once

While Loop - Example

```
int num=10;
while (num < 10)
{
    printf(“%d\n”, num);
    num = num +1;
}
```

Output
Nothing

Do While Loop - Example

```
int num=10;
do {
    printf(“%d\n”, num);
    num = num +1;
} while (num < 10);
```

Output
10

Break and Continue Keywords

- **break** Keyword

if the break statement is used in a loop body it allows you to exit from the current block as we saw in the switch statement. This can used to terminate the loop when an exceptional condition occurs.

- **continue** Keyword

if the continue keyword is used it takes you back to the condition in the loop. It allows you to restart the loop with next value when an exceptional condition occurs.

Break and Continue Examples

Break Keyword Example

```
int x = 3;
while (x <= 10)
{
    if (x == 7)
         break;
    printf("%d, ", x);
    x = x +1;
}
```

Output

```
3, 4, 5, 6,
```

Continue Keyword Example

```
for (x = 1; x<=10; x = x+1)
{
    if (x%3 == 0)
         continue;
    printf("%d, ", x);
}
```

Output

```
1, 2, 4, 5, 7, 8, 10
```

Infinite Loops

These are loops that never ends thus called infinite. These can be used effectively to serve a purpose by combining with break keyword. However, sometimes they do occur as a result of logic errors.

Infinite Loop – Good Example

```
while (1)  ← infinite loop
{
    char ch;
    scanf("%c", &ch);
    if (ch == '.')
        break;
    printf("%c", ch);
}
```

Output

```
Hello world
Hello world
David.
David
```

Infinite Loop – Logic Error (Bad)

```
int x = 1;
while (x <= 10)
{
    if (x%3 == 0)
        continue;
    printf("%d, ", x);
    x = x +1;
}
```

Output

```
1, 2,
```

Output

```
1, 2, 3, 3, 3, …
```

Lesson Summary

- If / Else Statement
- Nested IF
- Switch Statement
- While Loops
- For Loops
- Do While Loops
- Break and Continue Keywords
- Infinite Loops