Programming In C

Part IV

Control and Iterative Constructs

Flow of Control In C

- **Sequential**: Sequential execution of code statements (one line after another)
- **Decision**: used for decisions, branching -- choosing between 2 or more alternative paths. In C, these are the types of selection statements:
  - if
  - if/else
  - switch
- **Repetition**: used for looping, i.e. repeating a piece of code multiple times in a row. In C, there are three types of loops:
  - for
  - while
  - do/while

Decision with If

```
if (expresion){
statements
}
```

Decision with If-Else

```
if (expresion){
statements
}
else{
statements
}
```

Example 1

/*Compare two integers in C*/

```
#include <stdio.h>

int main() {
int a, b;

printf("Enter value for A :");
scanf("%d", &a);
printf("Enter value for B :");
scanf("%d", &b);

if(a > b)
  printf("a is greater than b");
else
  printf("a is not greater than b");

return 0;
}
```

Example 2

/*Compare three integers in C*/

```
#include <stdio.h>

int main() {
int a, b, c;

printf("Enter value for A :");
scanf("%d", &a);
printf("Enter value for B :");
scanf("%d", &b);
printf("Enter value for C :");
scanf("%d", &c);

if (a>b) {
   if(a>c)
        printf("a is greater than b and c");
   else
        printf("c is greater than a and b");
}
else {
   if(b>c)
        printf("b is greater than a and c ");
   else
        printf("c is greater than a and b ");
}

return 0;
}
```

Decision with switch (multiple-selection)

```
switch (integer expression) {
    case constant 1:
        statements;
        break;
    case constant 2:
        statements;
        break;
    .
    .
    .
    case constant n:
        statements;
        break;
    default:
        statements;
}
```

Example 3

```
#include<stdio.h>
int main( ) {
int a, b, c, choice;

printf("Enter two numbers\n");
scanf("%d%d", &a, &b);

printf("Enter your choice: 1 for addition and 2 for subtraction\n");
scanf("%d", &choice);

switch(choice) {
      case 1:
            c = a + b;
            printf("the sum is %d", c);
            break;
      case 2:
            c = a - b;
            printf("the sub is %d", c);
            break;
      default:
            printf("you have passed a wrong key");
}

return 0;
}
```

Iterative constructs looping

1. for
2. while
3. do/while

for loop

- This is one of the most frequently used loop in C programming.
- **Syntax of for loop:**

```
for (initialization; condition; increment/decrement) {
   statements;
 }
```

for loop - Example

```
/*Program to print first 10 natural
  numbers*/
#include<stdio.h>
int main( ) {
int x;

for(x = 1; x <= 10; x++) {
  printf("%d\n", x);
}

return 0;
}
```

Nested for loop

Syntax:

```
for(initialization; condition; increment/decrement) {
    for(initialization; condition; increment/decrement) {
        statements ;
    }
 }
```

Nested for loop – Example 1

```
/*Program to print square of stars*/
#include<stdio.h>
int main( ) {
int i, j;

for(i = 0; i < 5; i++) {
    printf("\n");
    for(j = 0; j <5; j++) {
        printf("*");
    }
}

return 0;
}
```

Output:

```
*****
*****
*****
*****
*****
```

Nested for loop – Example 2

```
/*Program to print half pyramid of stars*/
#include<stdio.h>
int main( ) {
int i, j;

for(i = 0; i < 5; i++) {
    printf("\n");
    for(j = 0; j <=i; j++) {
        printf("*");
    }
}

return 0;
}
```

Output:

```
*

**

***

****

*****
```

while loop

- **Syntax:**

```
while (condition) {
   statements;
}
```

while loop - Example

```
/*Program to print first 10 natural
  numbers*/
#include<stdio.h>
int main( ) {
int x;
x = 1;
while(x <= 10) {
  printf("%d\n", x);
  x++;
}
return 0;
}
```

do-while loop

- **Syntax:**

```
do {
statements;
}  while (condition)
```

do-while loop -Example

```
/*Program to print first 10 multiples
  of 5*/
#include<stdio.h>
int main() {
int a, i;
a = 5; i = 1;

do {
  printf("%d\t", a*i);
  i++;
} while(i <= 10);

return 0;
}
```