Control Statements

C Keywords

auto
break
case
char
const
continue
default
do

double
else
enum
extern
float
for
goto
if

int
long
register
return
short
signed
sizeof
static

struct
switch
typedef
union
unsigned
void
volatile
while

The words in **bold** print are used in control statements. They change the otherwise sequential execution of the assignment statements in a function block

Assignment Statement

```
aResult = 10;

aResult = countB;

aResult = (aValue + 100) / countD;

aResult = sqrt (aValue);
```

Block of Statements

```
{
const int MAX_BINS = 11;
int orderNumber;
int binNumber;

printf("Enter order number: ");
scanf("%d", &orderNumber);
binNumber = orderNumber % MAX_BINS;
printf("Bin number is %d\n", binNumber);
}
```

Simple “if” statement

```
if (aNumber != 1000)
   countA++;
```

“if” with a block of statements

```
if (aValue <= 10)
   {
   printf("Answer is %8.2f\n", aValue);
   countB++;
   } // End if
```

“if” – “else” with a block of statements

```
if (aValue <= 10)
   {
   printf("Answer is %8.2f\n", aValue);
   countB++;
   } // End if
else
   {
   printf("Error occurred\n");
   countC++;
   } // End else
```

Nested “if” statement

```
if (aValue == 1)
   countA++;
else if (aValue == 10)
   countB++;
else if (aValue == 100)
   countC++;
else
   countD++;
```

“while” with a single statement

```
while (aResult >= 1000)
   aResult = aResult / aValue;
```

“while” with a block of statements

```
while (aResult >= 1000)
   {
   aResult = aResult / aValue;
   printf("Result is %9.4f\n", aResult);
   } // End while
```

“while” with nested “if” statements

```
aResult = MAX_VALUE;

while (aResult >= 1000)
   {
   countW++;

   // nested if statement
   if (aValue == 1)
      countA++;
   else if (aValue == 10)
      countB++;
   else if (aValue == 100)
      countC++;
   else
      countD++;
   aResult = aResult / aValue;
   } // End while
```

“while” performing an infinite loop

```
while (1)
   {
   receiveFrom (aClient, aRequest);
   theResults = processRequest(aRequest);
   sendTo (aClient, theResults);
   } // End while
```

“for” with a single statement

```
for (i = 1; i <= MAX_LENGTH; i++)
   printf("#");
```

“for” with a block of statements

```
for (i = 0; i < MAX_SIZE; i++)
   {
   printf("Symbol is %c\n", aBuffer[i]);
   aResult = aResult / i;
   } // End for
```

“for” performing an infinite loop

```
for (;;)
   {
   receiveFrom (client, request);
   results = processRequest(request);
   sendTo (client, Results);
   } // End for
```

“switch” statement

```
switch (aNumber)
   {
   case 1   : countA++;
              break;
   case 10  : countB++;
              break;
   case 100 :
   case 500 : countC++;
              break;
   default  : countD++;
   } // End switch
```

☺

Jai Hind !! Jai Bangladesh !!