

Lesson 3

Loops

In this lesson, you will learn:

- About `do - while()`, `while()`, and `for()` loops
- How to use these loops
- When to use these loops

Important Notes:

- You should type in all the programs in this handout, and run them more than once with different data
- You should read, and understand everything in this handout, the material in it forms the basis of the quizzes
- If you don't understand something; ask me to explain.

The `do - while()` loop

The general form of the `do - while()` loop is:

```
do
{
    <program statements>
} while ( <test condition> );    // notice semi-colon here
```

Usage:

When you will loop at least once. This is a *conditional* loop, it will continue to loop as long as the test condition is true.

The `while()` loop

The general form of the `while()` loop is:

```
while ( <test condition> )
{
    <program statements>
}
```

Usage:

When you might not loop even once. This is another *conditional* loop, it will continue to loop as long as the test condition is true.

The `for()` loop

The general form of the `for()` loop is:

```
for ( <start assignment> ; <stop condition> ; <increment code> )
{
    <program statements>
}
```

Usage:

When you will loop a certain number of times. This loop is a *count* controlled loop.

These next three programs demonstrate how to use the different loops, to do the same thing.

The three programs will ask for 5 numbers from the user. The programs then total these numbers, and will calculate the average.

Program (total2.c)

Using a while() loop

```
/*
   Program "total2.c"
   Written by: Joe Dorward
   Date: 05/30/00

   Using a while() loop, his program asks the user for 5 numbers.
   The program totals these numbers, and calculates the average.
*/

#include <stdio.h>

void main(void)
{
    int loop_counter = 1,
        the_number = 0,
        the_total = 0,
        the_average = 0;

    while (loop_counter <= 5)
    {
        printf(" Please enter a number: ");
        scanf("%d",&the_number);

        the_total = the_total + the_number;
        loop_counter = loop_counter + 1;
    }

    the_average = the_total / 5;

    printf("\n The total is: %d \n",the_total);
    printf(" The average is: %d \n",the_average);
}
```

Program (total3.c)

Using a do - while() loop

```
/*
   Program "total3.c"
   Written by: Joe Dorward
   Date: 05/30/00

   Using a do - while() loop, his program asks the user for 5 numbers.
   The program totals these numbers, and calculates the average.
*/

#include <stdio.h>

void main(void)
{
    int loop_counter = 1,
        the_number = 0,
        the_total = 0,
        the_average = 0;

    do
    {
        printf(" Please enter a number: ");
        scanf("%d",&the_number);

        the_total = the_total + the_number;
        loop_counter = loop_counter + 1;

    } while(loop_counter <= 5);    // notice semi-colon here

    the_average = the_total / 5;

    printf("\n The total is: %d \n",the_total);
    printf(" The average is: %d \n",the_average);
}
```

Program (total4.c)

Using a for() loop

```
/*
   Program "total4.c"
   Written by: Joe Dorward
   Date: 05/30/00

   Using a for() loop, his program asks the user for 5 numbers.
   The program totals these numbers, and calculates the average.
*/

#include <stdio.h>

void main(void)
{
    int loop_counter = 1,
        the_number = 0,
        the_total = 0,
        the_average = 0;

    for (loop_counter = 1; loop_counter <= 5; loop_counter++)
    {
        printf(" Please enter a number: ");
        scanf("%d",&the_number);

        the_total = the_total + the_number;
    }

    the_average = the_total / 5;

    printf("\n The total is: %d \n",the_total);
    printf(" The average is: %d \n",the_average);
}
```

The next three programs demonstrate the most appropriate looping method for each situation.

Program (cars1.c)

```
/*
   Program "cars1.c"
   Written by: Joe Dorward
   Date: 05/30/00

   This is a toll-booth program.
   It asks the user to enter the number of passengers in
   each car that passes through a toll-booth.

   It keeps track of the number of cars, and passengers.
   When a negative number for passengers is entered, the
   program ends, and prints out the number of passengers
   and cars that have passed through the toll-booth.
*/

#include <stdio.h>

void main(void)
{
    int number_of_passengers = 0,
        total_passengers = 0,
        number_of_cars = 0;

    printf("How many passengers in this vehicle: ");
    scanf("%d", &number_of_passengers);

    while (number_of_passengers > 0)
    {
        number_of_cars++;    // increment car counter
        total_passengers = total_passengers + number_of_passengers;

        printf("How many passengers in this vehicle: ");
        scanf("%d", &number_of_passengers);
    }

    printf("\n There have been: %d vehicles.\n", number_of_cars);
    printf(" With a total of: %d passengers.\n", total_passengers);
}
```

Program (add_test2.c)

This program adds a “flag” to program add_test1.c, repeatedly asking for an answer until the user enters the correct answer.

```
/*
   Program "add_test2.c"
   written by: Joe Dorward
   Date: 05/30/00

   This program asks the user for two numbers.
   It then asks the user what they add up to.
   It then checks the answer in an if() statement,
   and prints out a right/wrong message.

   It will ask repeatedly for an answer until a correct
   one is entered
*/

#include <stdio.h>

void main(void)
{
    int first_number = 0,
        second_number = 0,
        the_answer = 0,
        flag = 0;    // flag for text condition

    // Ask for a number
    printf(" Please enter an integer: ");
    scanf("%d",&first_number);

    // Ask for a number
    printf(" Please enter an integer: ");
    scanf("%d",&second_number);

    do
    {
        // Ask the question
        printf(" +-----+");
        printf("\n What does %d + %d = ",first_number,second_number);
        scanf("%d",&the_answer);

        // Test the answer, and choose a message
        if (first_number + second_number == the_answer)
        {
            printf("\n Hey, you got it right! \n");
            flag = 1;
        }
        else
        {
            printf("\n Boy did you get it wrong! \n");
            printf(" Try again! \n");
        }
    }
```

```
    } while(flag == 0);  
}
```

Program (times2.c)

```
/*
   Program "times2.c"
   Written by: Joe Dorward
   Date: 03/19/00

   This program reads in an integer from the user,
   then writes out the multiplication-table for that integer.

   As a demonstration of the use of a for-loop, and field width specifiers.
*/

#include <stdio.h>

void main(void)
{
    int times = 0,
        number_of_loops = 0;

    printf("\nThe Multiplication Table Program\n");

    printf("\nPlease enter an integer: ");
    scanf("%d",&times);

    for (number_of_loops = 1; number_of_loops <= 12; number_of_loops++)
    {
        printf("\n%d times %2d = ",times,number_of_loops);
        printf("%3d",times * number_of_loops);
    }
}
```
