The **do ... while** loop

```c
int selection = 0;

do {
    printf("\n ");
    printf("\n1 - Add a Record ");
    printf("\n2 - Change a record ");
    printf("\n3 - Delete a record ");
    printf("\n4 - Quit ");
    printf("\n ");
    printf("\nEnter a selection: " );
    scanf("%d", &selection );
} while ( selection < 1 || selection > 4 );
```

Note that we have the same 3 parts as in a **for** loop, but written differently
do statements;
while (condition);
QUIZ

Write a do ... while loop that prints all the multiples of 3 between 50 and 100.

```c
int selection = 0;
do {
    printf("\n");
    printf("\n1 - Add a Record" );
    printf("\n2 - Change a record\n");
    printf("\n3 - Delete a record\n");
    printf("\n4 - Quit\n");
    printf("\n\n");
    printf("\nEnter a selection: " );
    scanf("%d", &selection );
}while ( selection < 1 || selection > 4 );
```
QUIZ

Write a do ... while loop that prints all the odd numbers between 25 and 42 in reverse order.
Important when working with loops

Check the boundary conditions (where the loop starts and where it ends) carefully.

Make sure you understand:

• What happens the first time through the loop
• What happens the last time through the loop
• How many times the body of the loop has executed
• What are the values of the variables after the loop has exited
What is the difference between these 2 types of loops?
To which of these 2 types does each of the 3 C loops belong?
How to choose the appropriate loop for the task at hand

First decision: test at the end or at the beginning?

• at end → **do ... while**

• at beginning:
  • Know # of iterations → **for**
  • Don’t know → **while**
Nested loops

The right way
Inner loop is completely inside the outer

The wrong way
Loops straddle each other
Your turn: What does this code print?

```c
int x, y, sum;
for (x = 1; x < 11; x++) {
    sum = 0;
    y = 1;
    do {
        sum += y++; // post-increment!
    } while (y <= x);
    printf("%i\t%i\n", x, sum);
}
```
Your turn: What does this code print?

```c
int x, y, sum;
for (x = 1; x < 11; x++) {
    sum = 0;
y = 1;
do {
    sum += y++; //post-increment!
} while (y <= x);
printf("%d\t%d\n", x, sum);
}
```
Your turn: What does this code print?

```c
int x, y, sum;
for (x = 1; x < 11; x++) {
    sum = 0;
    y = 1;
    do {
        sum += y++; //post-increment!
    } while (y < x);
    printf("%i\t%i\n", x, sum);
}
```
Your turn: What does this code print?

```c
int x, y, sum;
for (x = 1; x < 11; x++) {
    sum = 0;
y = 1;
do {
    sum += y++; // post-increment!
    } while (y < x);
printf("%d\t%d\n", x, sum);
}
```

```
1 1
2 1
3 3
4 6
5 10
6 15
7 21
8 28
9 36
10 45
```
Homework for Chapter 6:
• End-of-chapter Exercises 6, 7, 8, 9
• Due Friday, Oct. 7

For each coding exercise capture a screenshot of both code and output

Give instructor one printout with the entire homework.