

Programming Principles (MT162)

Lecture 8

Dr. Ahmed Fathalla

Some notes about the exam

Mistakes

- π is not a valid variable name.
- `if (ch==a) // 'a'`
 - `'=='` is used as a logical operator, not `'=`
 - char and String values must be enclosed between `' '` and `" "`, respectively.
- Expressions like `3.14r2` are not valid arithmetic expressions. `// 3.14*r*r`
 - Square can be calculated by `r*r`, or using ***pow*** function.
 - Multiplication operation is performed by using `*` operator only.
- A nice way of representing ***or*** logical operator in a switch case statement:
Case 'R':
Case 'r':
 `cout<<"red";break;`

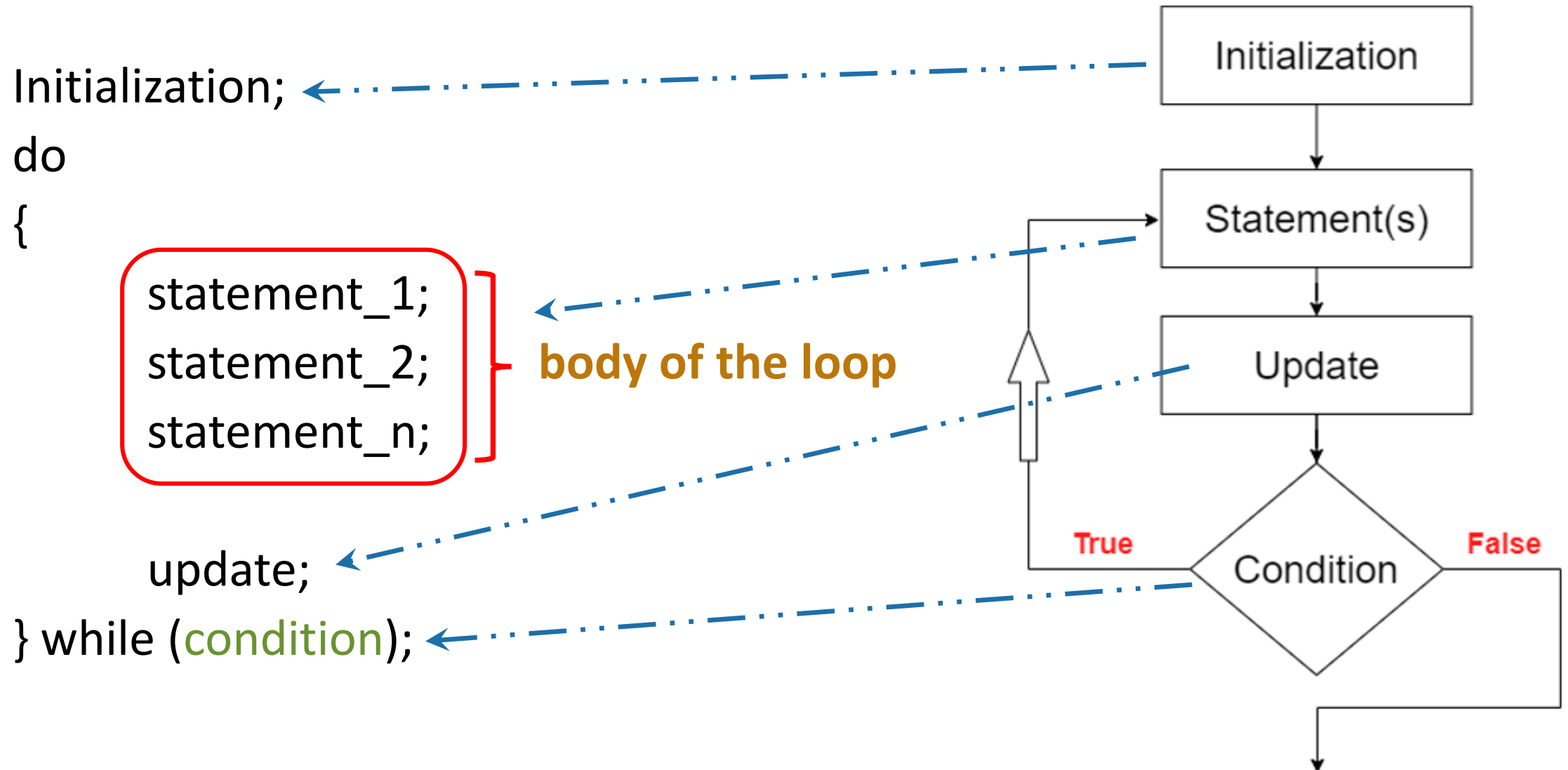
do...while Looping (Repetition) Structure

- General form of a do...while:

```
do
{
    statement
}
while (expression);
```

- The `statement` executes first, and then the `expression` is evaluated
- To avoid an infinite loop, body must contain a statement that makes the `expression` `false`
- The `statement` can be simple or compound
- Loop always iterates at least once

do-while loop

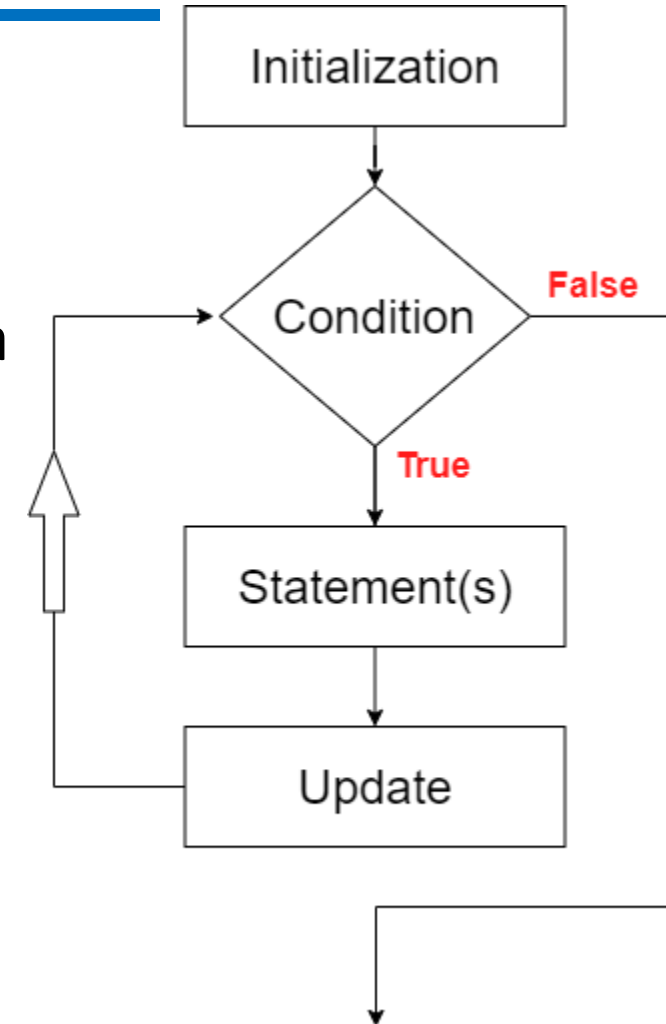


Exercise_1: write a program to sum input numbers until the user enters zero.

```
int main()
{
    double number, sum = 0;
    // the body of the loop is executed at least once
    do
    {
        cout<<"Enter a number: ";
        cin>>number;
        sum += number;
    } while(number != 0.0);
    cout<<"Sum = "<<sum;
    return 0;
}
```

Control Statement: break and continue

- The break statement, when executed it provides an immediate exit from the loop structure.
 - The break statement is typically used to exit early from a loop.
 - After the break statement executes, the program continues to execute with the first statement after the structure.
- The continue statement is used in while, for, and do.. while structures. When the continue statement is executed in a loop, it skips the remaining iteration/statements in the loop and proceeds with the next iteration of the loop.



Exercise_2: Print **even** numbers between [1, 100]. (using continue statement)

```
for(int i=1; i<=100; i++)  
{  
    if (i%2!=0)  
        continue;  
    cout<<i<<"\n";  
}
```

Exercise_3: Print numbers between [1, 100]. (using break statement)

```
for(int i=1; ; i++)  
{  
    if (i>100)  
    {  
        break;  
    }  
    cout<<i<<"\n";  
}
```

```
int i=1;  
while (true)  
{  
    if (i>100)  
        break;  
    cout<<i++<<"\n";  
}
```


Infinite loop

- **Infinite loop**: continues to execute endlessly
 - Avoided by including statements in loop body that assure exit condition is eventually `false`.
- Example of infinite loops:

```
for(  ; ;  ) {  
    cout<<"This loop will run forever.\n";  
}  
  
while(true)  
{  
    cout<<"This loop will run forever.\n";  
}
```

Exercise_4: Extend Exercise-1 to get average, min and max of the input numbers.

<https://onlinegdb.com/jzfvR6zwq>

```
#include <iostream>
```

```
#include <limits.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int number, sum = 0, count=0;
```

```
    int max=INT_MIN, min=INT_MAX;
```

```
    while (1)
```

```
    {
```

```
        cout<<"Enter a number: ";
```

```
        cin>>number;
```

```
        if (number==0)
```

```
            break;
```

```
        if (number>max)
```

```
            max=number;
```

```
        if (number<min)
```

```
            min=number;
```

```
        sum += number;
```

```
        count++;
```

```
    }
```

```
    cout<<"\n=====
\n";
```

```
    cout<<"Sum = "<<sum<<endl;
```

```
    cout<<"Avg = "<<sum/(float)count<<endl;
```

```
    cout<<"Min = "<<min<<endl;
```

```
    cout<<"Max = "<<max<<endl;
```

```
    return 0;
```

```
}
```

Exercise_4: Evaluate the following arithmetic expression.

$$\frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \dots + \frac{x^{100}}{100!}$$

Entry-controlled and Exit-controlled loops

- The loop in which test condition is checked in the beginning of the loop are known as **entry controlled loop**.
 - For Example: while & for loops.
- When statements inside the loop body is executed first and then the condition is checked that loop is known to be as **exit controlled loop**.
 - For Example: do-while loop.

Choosing the correct looping

- Number of repetitions is known —> **for loop**.
- Number of repetitions unknown + could be zero —> **while loop**.
- Number of repetitions unknown + at least 1 —> **do...while loop**.

Nested Control Structures

1. Write a c++ program to find the multiplication table of a given number
2. Modify the program to find the multiplication table of for all numbers between 1:10

Part_1

```
int n;  
cin>>n;  
for(int i=1; i<=n; i++)  
{  
    cout<<n*i<<"\t";  
}  
cout<<"\n";
```

Part_2

```
int n;  
for(n=1;n<=10;n++)  
{  
    for(int i=1; i<=n; i++)  
    {  
        cout<<n*i<<"\t";  
    }  
    cout<<"\n";  
}
```

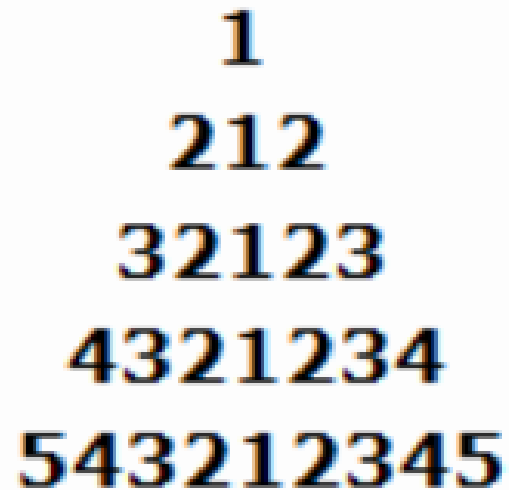
Nested Control Structures

Write a program to create the following pattern:

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

```
for (i = 1; i <= 5 ; i++)  
{  
    for (j = 1; j <= i; j++)  
        cout << "*";  
    cout << endl;  
}
```

Exercise_4: Write a program that prints the following pattern.



The pattern consists of five rows of numbers, centered and symmetric. The first row has one digit (1), the second has two (212), the third has three (32123), the fourth has four (4321234), and the fifth has five (543212345). The digits are bold and black, with a slight blue shadow effect.

```
1
212
32123
4321234
543212345
```


Solving *'s and numbers pattern problems

- https://www.youtube.com/playlist?list=PLwCMLs3sjOY4viWniHr0oMn0nyRU_G2dz