**Assignment No.1: Theory of Programming Languages**

Compare the structure of FOR loop in C++ and java and provide an analysis based on reliability writability and readability as discussed in chapter 1 of text book.

**Name: Muhammad Waqas Akhtar Wattoo**

**Reg. No: MSCS-3151**

**Submitted to: Sir. Abdul Hannan**

**Loop Introduction**

**Using Loops**

In Loop, the statement needs to be written only once and the loop will be executed 10 times as shown below.  
In computer programming, a loop is a sequence of instructions that is repeated until a certain condition is reached.

**For loop**

A for loop is a repetition control structure which allows us to write a loop that is executed a specific number of times. The loop enables us to perform n number of steps together in one line.  
**Syntax:**

for (initialization expr; test expr; update expr)

{

// body of the loop

// statements we want to execute

}

In for loop, a loop variable is used to control the loop. First initialize this loop variable to some value, then check whether this variable is less than or greater than counter value. If statement is true, then loop body is executed and loop variable gets updated . Steps are repeated till exit condition comes.

* **Initialization Expression**: In this expression we have to initialize the loop counter to some value. for example: int i=1;
* **Test Expression**: In this expression we have to test the condition. If the condition evaluates to true then we will execute the body of loop and go to update expression otherwise we will exit from the for loop. For example: i <= 10;
* **Update Expression**: After executing loop body this expression increments/decrements the loop variable by some value. for example: i++;

Equivalent flow diagram for loop:



**For Loop syntax/working in Java**

Java also includes another version of for loop introduced in Java 5. Enhanced for loop provides a simpler way to iterate through the elements of a collection or array. It is inflexible and should be used only when there is a need to iterate through the elements in sequential manner without knowing the index of currently processed element.  
Also note that the object/variable is immutable when enhanced for loop is used i.e it ensures that the values in the array can not be modified, so it can be said as read only loop where you can’t update the values as opposite to other loops where values can be modified.  
We recommend using this form of the for statement instead of the general form whenever possible.(as per JAVA doc.)

**Syntax:**

for (T element:Collection obj/array)

{

statement(s)

}

Lets take an example to demonstrate how enhanced for loop can be used to simpify the work. Suppose there is an array of names and we want to print all the names in that array. Let’s see the difference with these two examples  
Enhanced for loop simplifies the work as follows-

|  |
| --- |
| // Java program to illustrate enhanced for loop  public class enhancedforloop  {      public static void main(String args[])      {          String array[] = {"Ron", "Harry", "Hermoine"};            //enhanced for loop          for (String x:array)          {              System.out.println(x);          }            /\* for loop for same function          for (int i = 0; i < array.length; i++)          {              System.out.println(array[i]);          }          \*/      }  } |

Output:

Ron

Harry

Hermoine

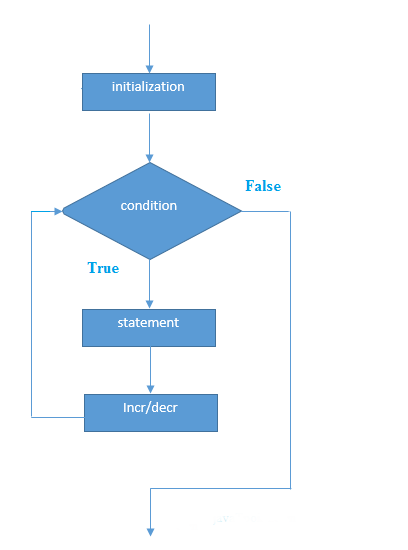
**Comparison of the structure of FOR loop in C++ and java**

A simple for loop is the same as C/C++ in Java . We can initialize the variable, check condition and increment/decrement value. It consists of four parts:

1. **Initialization**: It is the initial condition which is executed once when the loop starts. Here, we can initialize the variable, or we can use an already initialized variable. It is an optional condition.
2. **Condition**: It is the second condition which is executed each time to test the condition of the loop. It continues execution until the condition is false. It must return boolean value either true or false. It is an optional condition.
3. **Statement**: The statement of the loop is executed each time until the second condition is false.
4. **Increment/Decrement**: It increments or decrements the variable value. It is an optional condition.

**Syntax:**

1. **for**(initialization;condition;incr/decr){
2. //statement or code to be executed
3. }



**Example:**

1. //Java Program to demonstrate the example of for loop
2. //which prints table of 1
3. **public** **class** ForExample {
4. **public** **static** **void** main(String[] args) {
5. //Code of Java for loop
6. **for**(**int** i=1;i<=10;i++){
7. System.out.println(i);
8. }
9. }
10. }

**C++ for Loop**

Loops are used in programming to repeat a specific block of code. In this tutorial, you will learn to create a for loop in C++ programming

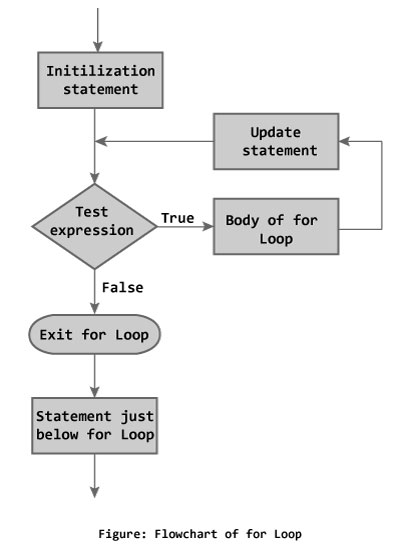
For Loop syntax/working in c++

for(initializationStatement; testExpression; updateStatement) {

// codes

}

where, only testExpression is mandatory.



## Example C++ for Loop

1. // C++ Program to find factorial of a number
2. // Factorial on n = 1\*2\*3\*...\*n
3. #include <iostream>
4. using namespace std;
5. int main()
6. {
7. int i, n, factorial = 1;
8. cout << "Enter a positive integer: ";
9. cin >> n;
10. for (i = 1; i <= n; ++i) {
11. factorial \*= i; // factorial = factorial \* i;
12. }
13. cout<< "Factorial of "<<n<<" = "<<factorial;
14. return 0;
15. }

We compare and examine languages for this purpose we will use Readability, Writability, Reliability as the base criteria.

**Readability**

Readability is directly related to the cost of maintenance. The different factors that impact readability are:

– Simplicity

– Control Statements

– Data types and data structures

– Syntax Considerations

– Orthogonality

***Simplicity***

A simple language will be easy to learn and hence understand. There are a number of factors that account for simplicity. These are:

• Number of basic components

Learning curve is directly proportional to the number of basic components. If a language has more basic components, it will be difficult to learn and vice-versa. It is important to note that one can learn a subset of a programming language for writing but for reading you must know everything.

• Feature multiplicity

If a language has more than one way to accomplish the same task, then it can cause confusion and complexity. For example, a number can be incremented in C in any of the following manners.

**i = i + 1;**

**i++;**

**++i;**

**i += 1;**

***Control Statements***

Control statements also play an important role in readability. We are all aware of the hazards of goto statement. If a language relies on goto statements for control flow, the logic becomes difficult to follows and hence understand and maintain. Restricted use of goto in extreme was needed and useful as well but with the emergence of new language constructs to handle those situations, it probably not an issue any more.

***Data types and data structures***

Data types and data structures also play an important role in improving the abstraction level of a programming language, hence making it more readable. For example, a simple omission of Boolean type in C resulted in many problems as integers were instead of Boolean which made the program difficult to follow and debug in many cases.

***Syntax***

Syntax of a language plays a very important role in programming languages. There are many different considerations that we will discuss throughout this course. Here we give a brief overview of some of these considerations.

• Variable names

The first consideration is the restriction of variable names. In the beginning it was a huge issue as some languages restricted the length of a variable name up to 6 or

8 characters making it difficult to choose meaningful names for the variable. Other considerations include the type of special characters allowed in the variable names to improve readability.

• Special keywords for signaling the start and end of key words.

**Writability**

Writability is a proportion of help for deliberation gave by a language. It is the capacity to characterize and utilize convoluted structures without making a big deal about the subtleties. It might be noticed that the level of reflection is legitimately identified with the expressiveness. For instance, on the off chance that you are approached to execute a tree, it will be a great deal troublesome in JAVA when contrasted with C++. It likewise is a proportion of expressivity: develops that make it advantageous to compose programs.

Symmetry

Symmetry is a significant idea. It tends to how moderately modest number of parts that can be joined in a generally modest number of approaches to get the ideal outcomes. It is intently connected with straightforwardness: the more symmetrical the structure the less exemptions and it makes it simpler to learn, read, and compose programs in a programming language. The significance of a symmetrical element is autonomous of the unique circumstance. The key parameters are balance and consistency. For instance pointers is a symmetrical idea.

**Reliability**

Unwavering quality is one more significant factor. A programming language should empower the software engineers to compose solid code. The significant characteristics that assume a significant job in this Unwavering quality is one progressively critical factor. A programming language ought to engage the product specialists to form strong code. The noteworthy characteristics that accept a huge activity in such manner are recorded underneath:

• Type Checking

Type checking is associated with checking the sort of the components used as operands in different. The request that ought to be tended to is whether this sort checking is done at consolidate time or at run-time. This moreover joins checking sorts of parameters and bunch limits. Usually, these two have been the huge wellsprings of bungles in undertakings which are incredibly difficult to research. A language that types checking at mastermind time creates more trustworthy code than the one that does it at run-time.

• Exception managing

Exception dealing with enables a product architect to catch run-time bumbles and take medicinal measure if possible and subsequently making it possible to create dynamically reliable code.

**Conclusion:**

C++ is object-oriented programming (OOP) language

Java is programming language developed by Sun Microsystems

Bjarne Stroustrup developed C++ at AT & T Bell Laboratories in Murray Hil.

The Java language was initially called OAK. Originally, it was developed for handling devices and set-top boxes.

No strict relationship between class names and filenames. In C++, header files and implementation files are used for specific class.

The strict relationship is enforced, e.g., the source code for class PayRoll has to be in PayRoll.java