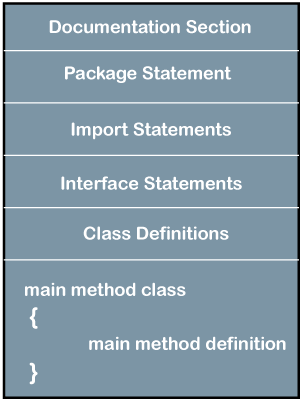
# Structure of Java Program

Java is an object-oriented programming, platform-independent, and secure programming language that makes it popular. Using the Java programming language, we can develop a wide variety of applications. So, before diving in depth, it is necessary to understand the basic structure of Java program in detail. In this section, we have discussed the basic structure of a Java program. At the end of this section, you will able to develop the Hello world Java program, easily.



Let's see which elements are included in the structure of a Java program. A typical structure of a Java program contains the following elements:

## Documentation Section

The documentation section is an important section but optional for a Java program. It includes basic information about a Java program. The information includes the author's name, date of creation, version, program name, company name, and description of the program. It improves the readability of the program. Whatever we write in the documentation section, the Java compiler ignores the statements during the execution of the program. To write the statements in the documentation section, we use **comments**. The comments may be **single-line, multi-line,** and **documentation** comments.

* **Single-line Comment:** It starts with a pair of forwarding slash **(//)**. For example:

1. //First Java Program

* **Multi-line Comment:** It starts with a **/\*** and ends with **\*/.** We write between these two symbols. For example:

1. /\*It is an example of
2. multiline comment\*/

* **Documentation Comment:** It starts with the delimiter **(/\*\*)** and ends with **\*/**. For example:

1. /\*\*It is an example of documentation comment\*/

## Package Declaration

The package declaration is optional. It is placed just after the documentation section. In this section, we declare the **package name** in which the class is placed. Note that there can be **only one package** statement in a Java program. It must be defined before any class and interface declaration. It is necessary because a Java class can be placed in different packages and directories based on the module they are used. For all these classes package belongs to a single parent directory. We use the keyword **package** to declare the package name. For example:

## Documentation Section

The documentation section is an important section but optional for a Java program. It includes **basic information** about a Java program. The information includes the **author's name, date of creation, version, program name, company name,** and **description** of the program. It improves the readability of the program. Whatever we write in the documentation section, the Java compiler ignores the statements during the execution of the program. To write the statements in the documentation section, we use **comments**. The comments may be **single-line, multi-line,** and **documentation** comments.

* **Single-line Comment:** It starts with a pair of forwarding slash **(//)**. For example:

1. //First Java Program

* **Multi-line Comment:** It starts with a **/\*** and ends with **\*/.** We write between these two symbols. For example:

1. /\*It is an example of
2. multiline comment\*/

* **Documentation Comment:** It starts with the delimiter **(/\*\*)** and ends with **\*/**. For example:

1. /\*\*It is an example of documentation comment\*/

## Package Declaration

The package declaration is optional. It is placed just after the documentation section. In this section, we declare the **package name** in which the class is placed. Note that there can be **only one package** statement in a Java program. It must be defined before any class and interface declaration. It is necessary because a Java class can be placed in different packages and directories based on the module they are used. For all these classes package belongs to a single parent directory. We use the keyword **package** to declare the package name. For example:

1. public class Demo //class definition
2. {
3. public static void main(String args[])
4. {
5. void display()
6. {
7. System.out.println("Welcome to javatpoint");
8. }
9. //statements
10. }
11. }

When we follow and use the above elements in a Java program, the program looks like the following.

1. /\*Program name: Palindrome\*/
2. //Author's name: Mathew
3. /\*Palindrome is number or string that will remains the same
4. When we write that in reverse order. Some example of
5. palindrome is 393, 010, madam, etc.\*/
6. //imports the Scanner class of the java.util package
7. import java.util.Scanner;
8. //class definition
9. public class CheckPalindromeNumber
10. {
11. //main method
12. public static void main(String args[])
13. {
14. //variables to be used in program
15. int r, s=0, temp;
16. int x; //It is the number variable to be checked for palindrome
17. Scanner sc=new Scanner(System.in);
18. System.out.println("Enter the number to check: ");
19. //reading a number from the user
20. x=sc.nextInt();
21. //logic to check if the number id palindrome or not
22. temp=x;
23. while(x>0)
24. {
25. r=x%10;  //finds remainder
26. s=(s\*10)+r;
27. x=x/10;
28. }
29. if(temp==s)
30. System.out.println("The given number is palindrome.");
31. else
32. System.out.println("The given number is not palindrome.");
33. }
34. }