# Java 9 Try With Resource Enhancement



Java introduced **try-with-resource** feature in Java 7 that helps to close resource automatically after being used.

In other words, we can say that we don't need to close resources (file, connection, network etc) explicitly, try-with-resource close that automatically by using AutoClosable interface.

In Java 7, try-with-resources has a limitation that requires resource to declare locally within its block.

**Example Java 7 Resource Declared within resource block**

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
public class FinalVariable {
   public static void main(String[] args) throws FileNotFoundException {
      try(FileOutputStream fileStream=new FileOutputStream("javatpoint.txt");){
          String greeting = "Welcome to javaTpoint.";
            byte b[] = greeting.getBytes();
            fileStream.write(b);
            System.out.println("File written");
      }catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

This code executes fine with Java 7 and even with Java 9 because Java maintains it's legacy.

But the below program would not work with Java 7 because **we can't put resource declared outside the try-with-resource.**

## Java 7 Resource declared outside the resource block

If we do like the following code in Java 7, compiler generates an error message.

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
public class FinalVariable {
   public static void main(String[] args) throws FileNotFoundException {
      FileOutputStream fileStream=new FileOutputStream("javatpoint.txt");
      try(fileStream){
          String greeting = "Welcome to javaTpoint.";
            byte b[] = greeting.getBytes();
            fileStream.write(b);
            System.out.println("File written");
      }catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

Output:

```
error: <identifier> expected
                   try(fileStream){
```

To deal with this error, try-with-resource is improved in Java 9 and now we can use reference of the resource that is not declared locally.

In this case, **if we execute the above program using Java 9 compiler, it will execute nicely without any compile error.**

## Java 9 try-with-resource Example

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
public class FinalVariable {
   public static void main(String[] args) throws FileNotFoundException {
      FileOutputStream fileStream=new FileOutputStream("javatpoint.txt");
      try(fileStream){
          String greeting = "Welcome to javaTpoint.";
            byte b[] = greeting.getBytes();
            fileStream.write(b);
            System.out.println("File written");
      }catch(Exception e) {
         System.out.println(e);
      }
   }
}
```

Output:

```
File written
```