```
PS C:\Users\ranga\OneDrive\Desktop\JavaCore\Class-5-Packages-Static-import-Abstract-Classes-Gen
erics-and-Collections> javac Main.java
Main.java:7: error: non-static variable PI cannot be referenced from a static context
        System.out.println(A.PI);
                           ^
1 error
PS C:\Users\ranga\OneDrive\Desktop\JavaCore\Class-5-Packages-Static-import-Abstract-Classes-Gen
erics-and-Collections>
```

Static Import

Gives a benefit over the ordinary import. In this import, we can directly import the variable/constant/method of the class directly so that we can use that particular variable/constant/method without mentioning the name of the class.

Using extends and implements keywords:

Using **extends** keyword in inheritance in Java:

1. We can extend a class from another class.
    a. E.g., public class B extends A
2. We can extend an interface using the **extend** keyword from another interface.
    a. E.g., public interface ChildInt extends ParentI

Using **implements** keyword in inheritance in Java

1. If a class inherits an interface, we use the keyword implements
2. ~~If an interface extends a class then~~ **X** is Not a Valid case scenario

Abstract classes

1. First level of development over an interface.

Definition:

If a class has at least one abstract method, then we need to declare that method with an abstract keyword and also the class.

Definition

An abstract method is a method that will have only the method **signature**. It won't have a method body.

e.g. of method signature: public int add(int num1, int num2); // Method signature

compensatory: public abstract int add(int num1, int num2); // Method signature

Interfaces

1. All methods should be abstract.
2. All variables/constants will automatically become public static and final
    a. e.g. int a = 10;
        i. After compilation it will become as:
            public static final int a = 10;
3. All methods in the interface are inherently abstract.
4. All variables will be inherently public static and final.