**Assisted Practice: 2.5 Inner Classes**

This section will guide you to:

* Create a Java project in your IDE
* Write a Java program to create inner classes

This lab has three subsections, namely:

* + 1. Writing a program in Java to verify the implementation of inner classes
    2. Executing the program and verifying working of inner classes
    3. Pushing the code to your GitHub repositories

**Step 2.5.1:** Writing a program in Java to verify the implementation of inner classes

There are two ways you can perform this step; you can create a new Java project, or you can create a new Java class in the existing project. It is preferable to create a new Java class in the existing project but feel free to explore the first option. The steps mentioned below will work once you create a project in Java.

* Open Eclipse
* *[Right click]* on the **src** folder of the project
* Select *New* -> *Java Class* -> Enter the filename (follow camelCasing)
* Execute the code below resolving the warning and errors due compatibility-related issues

**public** **class** innerClassAssisted1 {

**private** String msg="Welcome to Java";

**class** Inner{

**void** hello(){System.***out***.println(msg+", Let us start learning Inner Classes");}

}

**public** **static** **void** main(String[] args) {

innerClassAssisted1 obj=**new** innerClassAssisted1();

innerClassAssisted1.Inner in=obj.**new** Inner();

in.hello();

}

}

**public** **class** innerClassAssisted2 {

**private** String msg="Inner Classes";

**void** display(){

**class** Inner{

**void** msg(){

System.***out***.println(msg);

}

}

Inner l=**new** Inner();

l.msg();

}

**public** **static** **void** main(String[] args) {

innerClassAssisted2 ob=**new** innerClassAssisted2 ();

ob.display();

}

}

//anonymous inner class

**abstract** **class** AnonymousInnerClass {

**public** **abstract** **void** display();

}

**public** **class** innerClassAssisted3 {

**public** **static** **void** main(String[] args) {

AnonymousInnerClass i = **new** AnonymousInnerClass() {

**public** **void** display() {

System.***out***.println("Anonymous Inner Class");

}

};

i.display();

}

}

**Step 2.5.2:** Executing the program and verifying the working of inner classes

Before you execute the program, check for syntactical corrections. If no errors are found, follow the steps mentioned below:

* ***[Right click]*** in the program space
* Select *Run As Java Application*







**Step 2.5.3:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**