***Lab 1***

***Assignment-1***

* Create a class ***Student*** in ***Student.java*** then add member variables studentName, collegeName of type String
* Add a member variable studentID of type int.
* Make all the member variables as private.
* Add a main method. And print a message “Successful”.
* Compile the class
* Run the class (Follow Coding convention)

***Program***

**public** **class** Student {

**private** *String* studentName, collegeName;

**private** *int* studentID;

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

        System.out.println("Successful.");

    }

}

***Output***



***Assignment-2***

* Create a new class **Employee**
* Add member variables: id and age of type int, name of type String and isPermanent of type Boolean
* Now assign values 35.5 to age; See the error message.
* How can you avoid this error? Correct the error by casting.
* Make all the members protected
* Add a main method to it. Print message “Successfully started”.
* Compile the class.

***Program***

**public** **class** Employee {

**protected** *int* id, age **=** (*int*) 35.5;

**protected** *String* name;

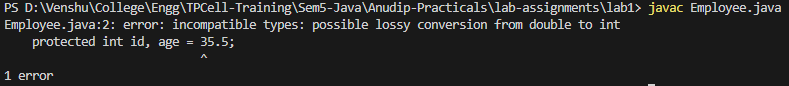
**protected** *boolean* isPermanent;

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

        System.out.println("Successful");

    }

}

***Output with error***

***Output without error***



***Assignment-3***

* Create a class **Person**
* Add member variables name as String, age and salary as int
* Initialize the member variable along with declaration.
* Now put the previous Person class in a package **com.anudip.learning**
* Add a main method. Add a print message “Test Successful”.
* Run the class after compilation.
* Modify the classpaths to see the error messages on the console.

***Program***

**public** **class** Person {

**private** *String* name **=** "Venshu";

**private** *int* age **=** 20, salary **=** 1500000;

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

        System.out.println("Test Successful.");

    }

}

***Output***

***Before changing classpaths***



***After changing classpaths***

