 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)

SOURCE CODE :

public class Student {

    private String studentname ;

    private String collegename ;

    private int studentID ;

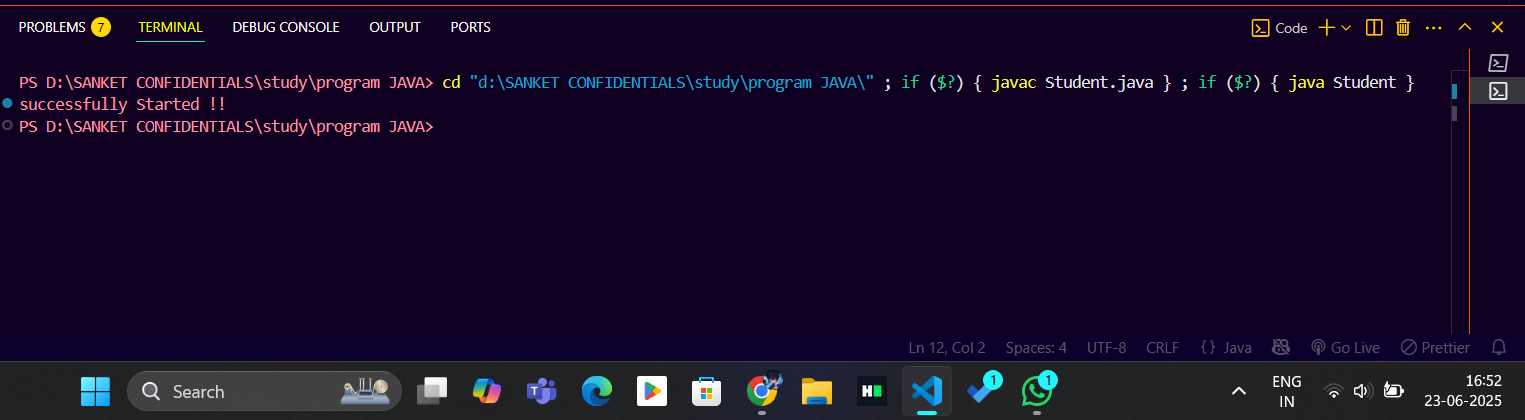
    public static void main(String[] args) {

        System.out.println("successfully Started !!");

    }

}

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.

SOURCE CODE :

public class Employee {

    protected int id,age;

    protected String name;

    protected boolean ispermanent;

    public static void main(String[] args) {

        Employee em1 = new Employee();

        em1.id = 100;

*em1.age = 35.5;   //datatype mismatch*

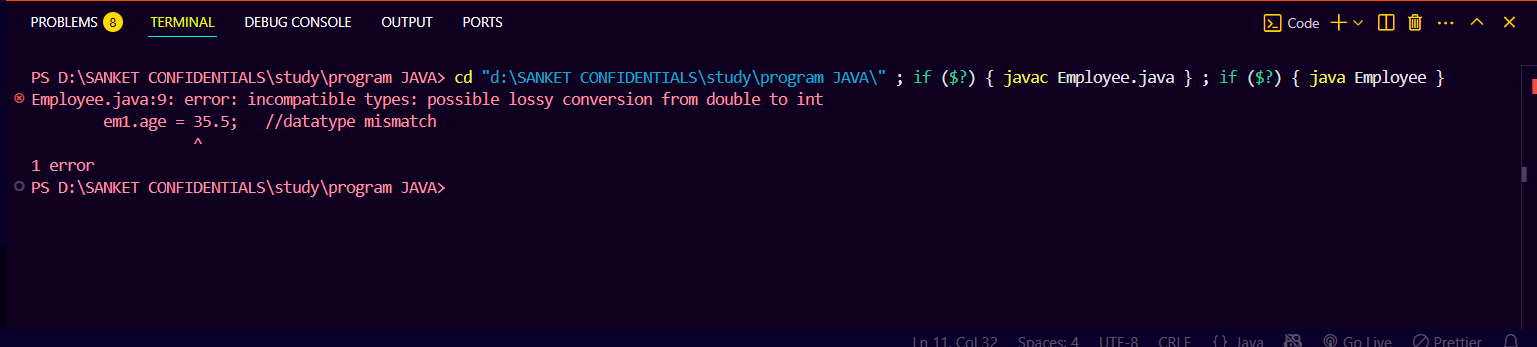
        em1.ispermanent = true;

        System.out.println("successfully started !!");

    }

}

OUTPUT : (ERROR – MISMATCH DATATYPE)



SOURCE CODE - CORRECTED

public class Employee {

    protected int id,age;

    protected String name;

    protected boolean ispermanent;

    public static void main(String[] args) {

        Employee em1 = new Employee();

        em1.id = 100;

*// em1.age = 35.5;   //datatype mismatch*

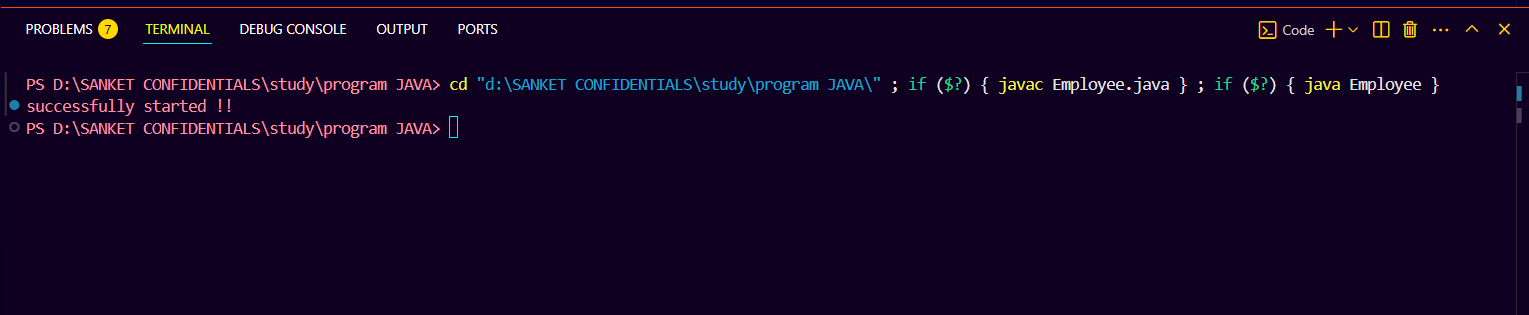
        em1.age = (int)35.5; //implicit type cast

        em1.ispermanent = true;

        System.out.println("successfully started !!");

    }

}



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.

SOURCE CODE :

package com.anudip.learning;

public class Person {

    String name = "ganesh";

    int age = 25 ;

    int salary = 10000;

    public static void main(String[] args) {

        System.out.println("Test successfull !!");

    }

}

OUTPUT :

