

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 :



The screenshot shows a terminal window with the following commands and output:

```
PS D:\SANKET CONFIDENTIALS\study\program JAVA> cd "d:\SANKET CONFIDENTIALS\study\program JAVA\" ; if ($?) { javac Student.java } ; if ($?) { java Student }  
● successfully Started !!  
PS D:\SANKET CONFIDENTIALS\study\program JAVA>
```

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)

A screenshot of a terminal window with a dark background. The terminal shows the command prompt 'PS D:\SANKET CONFIDENTIALS\study\program JAVA>' followed by the command 'cd "d:\SANKET CONFIDENTIALS\study\program JAVA\" ; if (\$?) { javac Employee.java } ; if (\$?) { java Employee }'. Below the command, an error message is displayed: 'Employee.java:9: error: incompatible types: possible lossy conversion from double to int' followed by the code snippet 'em1.age = 35.5; //datatype mismatch' with a red squiggly line under '35.5'. At the bottom, it says '1 error' and 'PS D:\SANKET CONFIDENTIALS\study\program JAVA>'. The terminal window has tabs for 'PROBLEMS', 'TERMINAL', 'DEBUG CONSOLE', 'OUTPUT', and 'PORTS' at the top. The 'TERMINAL' tab is active. There are also icons for 'Code', a search icon, a refresh icon, and a close icon in the top right corner of the terminal window.

```
PS D:\SANKET CONFIDENTIALS\study\program JAVA> cd "d:\SANKET CONFIDENTIALS\study\program JAVA\" ; if ($?) { javac Employee.java } ; if ($?) { java Employee }
Employee.java:9: error: incompatible types: possible lossy conversion from double to int
    em1.age = 35.5; //datatype mismatch
               ^
1 error
PS D:\SANKET CONFIDENTIALS\study\program JAVA>
```

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 !!");  
    }  
}
```



The screenshot shows a terminal window with the following content:

```
PS D:\SANKET CONFIDENTIALS\study\program JAVA> cd "d:\SANKET CONFIDENTIALS\study\program JAVA\" ; if ($?) { javac Employee.java } ; if ($?) { java Employee }  
● successfully started !!  
PS D:\SANKET CONFIDENTIALS\study\program JAVA>
```

The terminal window has a dark background with a light blue border. The title bar at the top includes tabs for PROBLEMS, TERMINAL, DEBUG CONSOLE, OUTPUT, and PORTS. The terminal output shows the command to compile and run the Java program, followed by the successful execution message.

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.

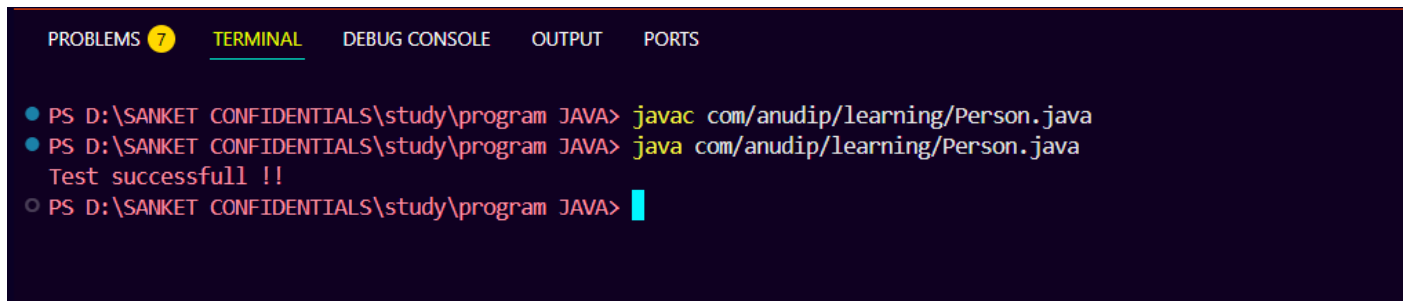
SOURCE CODE :

```
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 :

A screenshot of an IDE terminal window. At the top, there are tabs: 'PROBLEMS' with a yellow circle containing the number 7, 'TERMINAL' (which is selected and underlined), 'DEBUG CONSOLE', 'OUTPUT', and 'PORTS'. The terminal shows three lines of command history. The first line is 'PS D:\SANKET CONFIDENTIALS\study\program JAVA> javac com/anudip/learning/Person.java'. The second line is 'PS D:\SANKET CONFIDENTIALS\study\program JAVA> java com/anudip/learning/Person.java', followed by the output 'Test successfull !!'. The third line is 'PS D:\SANKET CONFIDENTIALS\study\program JAVA>' with a blue cursor at the end.

```
PROBLEMS 7 TERMINAL DEBUG CONSOLE OUTPUT PORTS

● PS D:\SANKET CONFIDENTIALS\study\program JAVA> javac com/anudip/learning/Person.java
● PS D:\SANKET CONFIDENTIALS\study\program JAVA> java com/anudip/learning/Person.java
  Test successfull !!
○ PS D:\SANKET CONFIDENTIALS\study\program JAVA> █
```