Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**8**
Student Name:	**RAVEENA PITALE**
Roll No :	**27**

Title:

1. To perform Multilevel Inheritance in JAVA. Create a Person class representing name, age and address. Inherit person class to employee class with emp ID and salary factor. Inherit the Employee class to programmer class with technical skills and hike attributes. Implement valid methods to input the details from the user in the main method and display for 3 programmers.
2. To perform Hierarchical Inheritance in JAVA. Create an Employee class with attributes EmpID and EmpSalary. Also create necessary methods/constructors to accept these values from the user. Create classes permenantEmployee and TemporaryEmployee which will be derived classes of Employee. Mention hike attribute in these derived classes and calculate the total salary using generate_salary() method for respective types of employees. Objects of the derived classes should be created and salaries for the permanent and temporary employees should be calculated and displayed on the screen.

Learning Objective:

- Students will be able to perform multilevel inheritance using JAVA.
- Students will be able to perform hierarchical inheritance using JAVA

Learning Outcome:

- To understand how to use the private members using friend function and friend class.

Course Outcome:

ECL304.2	Comprehend building blocks of OOPs language, inheritance, package and interfaces.

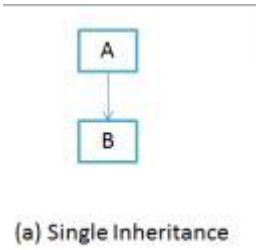
Faculty: Ms. Deepali Kayande

Theory:

- Explain in details about various inheritance types supported in JAVA

1) Single Inheritance

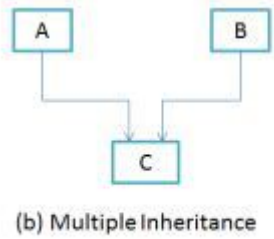
Single inheritance is damn easy to understand. When a class extends another one class only then we call it a single inheritance. The below flow diagram shows that class B extends only one class which is A. Here A is a **parent class** of B and B would be a **child class** of A.



(a) Single Inheritance

2) Multiple Inheritance

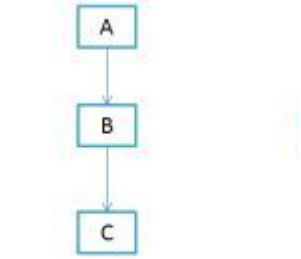
"**Multiple Inheritance**" refers to the concept of one class extending (Or inherits) more than one base class. The inheritance we learnt earlier had the concept of one base class or parent. The problem with "multiple inheritance" is that the derived class will have to manage the dependency on two base classes.



(b) Multiple Inheritance

3) Multilevel Inheritance

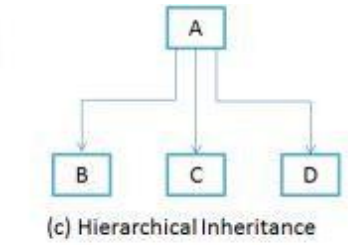
Multilevel inheritance refers to a mechanism in OO technology where one can inherit from a derived class, thereby making this derived class the base class for the new class. As you can see in below flow diagram C is subclass or child class of B and B is a child class of A.



(d) Multilevel Inheritance

4) Hierarchical Inheritance

In such kind of inheritance one class is inherited by many **sub classes**. In below example class B,C and D **inherits** the same class A. A is **parent class (or base class)** of B,C & D.

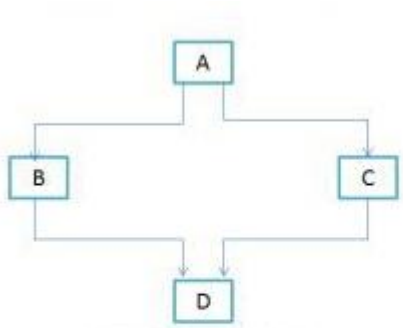


(c) Hierarchical Inheritance

5) Hybrid Inheritance

In simple terms you can say that Hybrid inheritance is a combination of **Single** and **Multiple inheritance.** A typical flow diagram would look like below. A hybrid inheritance can be achieved in the java in a same way as multiple inheritance can be!! Using interfaces. yes you heard it right. By using **interfaces** you can have multiple as well as **hybrid inheritance** in Java.

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22



(e) Hybrid Inheritance

Algorithm :01	1 – Create a parent class person and initialize its data members and take input of name, age and address. 2- Create the derived class of person class - employee class to take input of emp_id and salaryfactor. 3- Create the derived class of person class - programmer class to take input of hike and technical skills. 4 – Create the Main class to call the class functionalities and display the results.

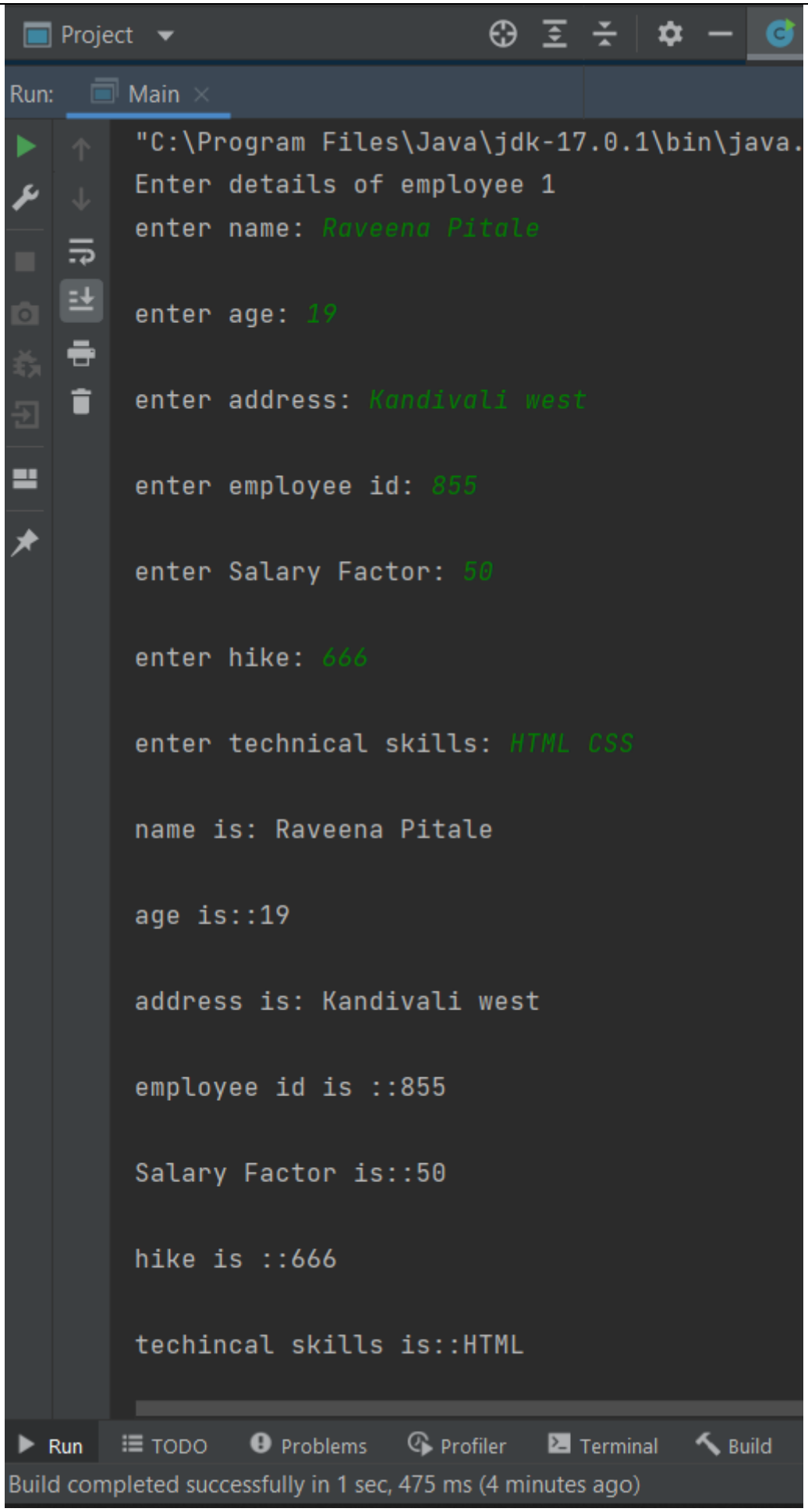
Program:01	`package com.company;`
	`import java.util.*;`
	
	`class Person {`
	
	`String name;`
	`int age;`
	`String address;`
	
	`public Person() {`
	`name = "";`
	`age = 0;`
	`address = "";`
	`}`
	
	`void getdata() {`
	`Scanner s = new Scanner(System.in);`
	`System.out.print("enter name: ");`
	`name = s.nextLine();`
	`System.out.println();`
	`System.out.print("enter age: ");`
	`age = s.nextInt();`
	`System.out.println();`
	`System.out.print("enter address: ");`
	`s.nextLine();`

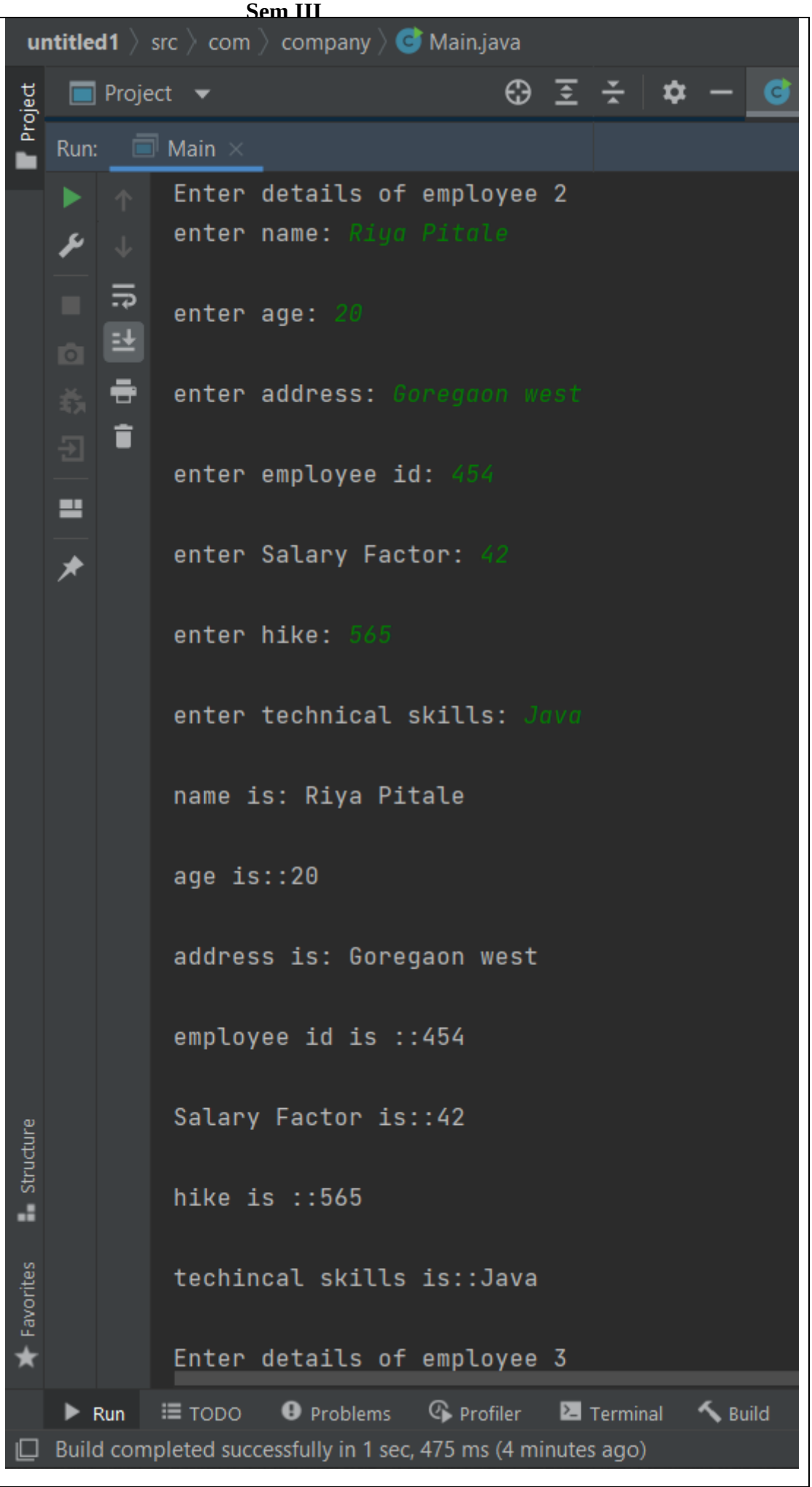
Faculty: Ms. Deepali Kayande

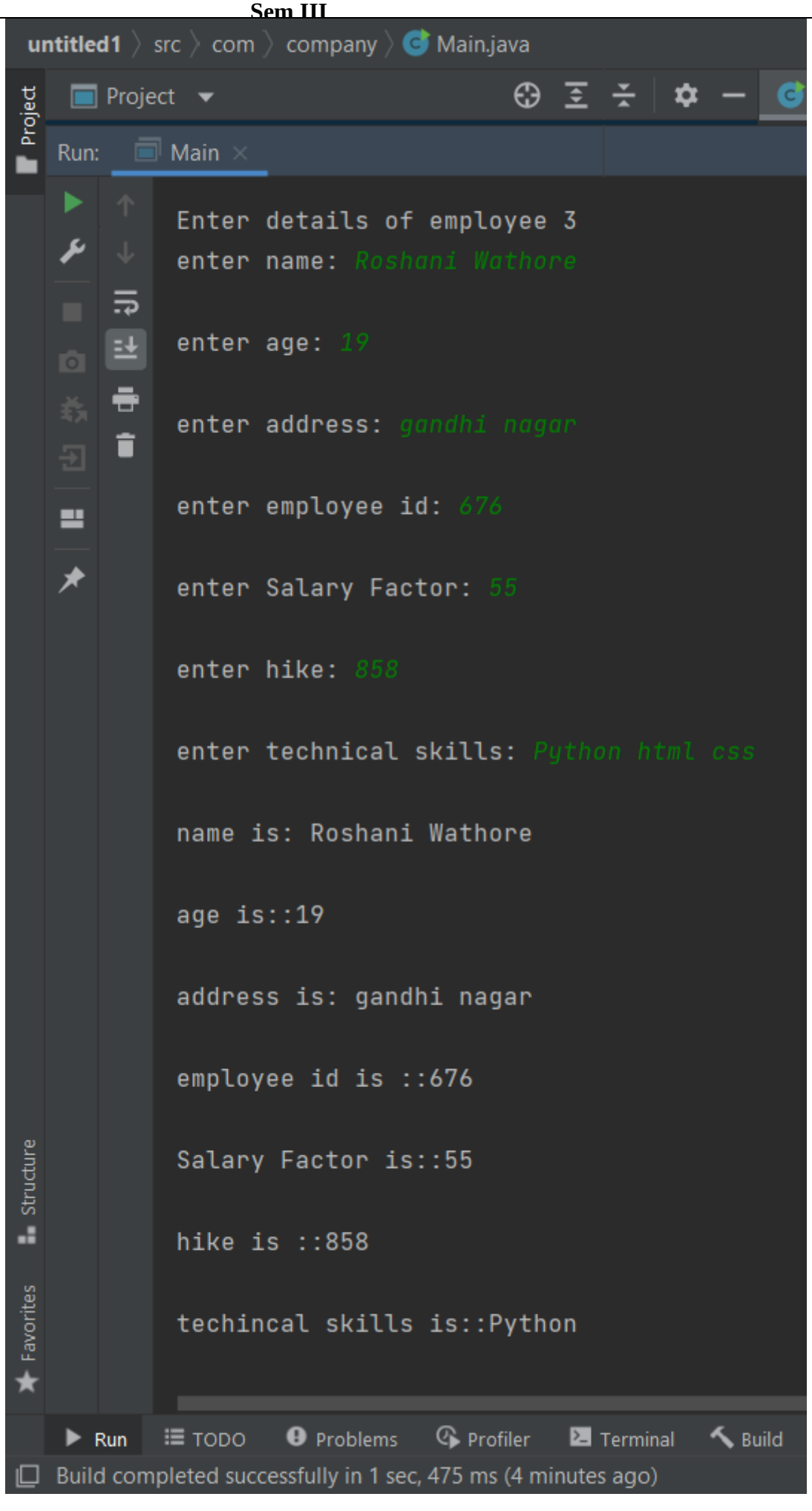
```
  address = s.nextLine();
  System.out.println();
 }
 void putdata() {
  System.out.println("name is: " + name);
  System.out.println();
  System.out.println("age is::" + age);
  System.out.println();
  System.out.println("address is: " + address);
  System.out.println();
 }
}
class employee extends Person {
 int emp_id;
 int salary_factor;

 public employee() {
  emp_id = 0;
  salary_factor = 0;

 }
 void getdetails() {
  Scanner s = new Scanner(System.in);
  System.out.print("enter employee id: ");
  emp_id = s.nextInt();
  System.out.println();
  System.out.print("enter Salary Factor: ");
  salary_factor = s.nextInt();
  System.out.println();

 }
 void putdetails() {
  System.out.println("employee id is ::" + emp_id);
  System.out.println();
  System.out.println("Salary Factor is::" + salary_factor);
  System.out.println();
 }
}
class programmer extends employee {
 int hike;
 String technical_skills = "";

 public programmer() {
  hike = 0;
  technical_skills = "";

 }
 void getd() {
  Scanner s = new Scanner(System.in);
  System.out.print("enter hike: ");
  hike = s.nextInt();
  System.out.println();
  System.out.print("enter technical skills: ");
  s.nextLine();
  technical_skills = s.next();
  System.out.println();
 }
```

```
 void putd() {
  System.out.println("hike is ::" + hike);
  System.out.println();
  System.out.println("techincal skills is::" + technical_skills);
  System.out.println();
 }
}


public class Main {

 public static void main(String[] args) {
  programmer r[] = new programmer[4];
  r[0] = new programmer();
  r[1] = new programmer();
  r[2] = new programmer();

  for (int i = 0; i<3; i++) {
   System.out.println("Enter details of employee " + (i+1)  );

   r[i].getdata();
   r[i].getdetails();
   r[i].getd();
   r[i].putdata();
   r[i].putdetails();
   r[i].putd();
  }
 }
}
```

Output Screenshot:01




untitled1 src com company Main.java
Project
Run: Main
Enter details of employee 2
enter name: Riya Pitale
enter age: 20
enter address: Goregaon west
enter employee id: 454
enter Salary Factor: 42
enter hike: 565
enter technical skills: Java
name is: Riya Pitale
age is::20
address is: Goregaon west
employee id is ::454
Salary Factor is::42
hike is ::565
techincal skills is::Java
Enter details of employee 3
Run TODO Problems Profiler Terminal Build
Build completed successfully in 1 sec, 475 ms (4 minutes ago)
Structure
Favorites


untitled1 src com company Main.java
Project
Run: Main
Enter details of employee 3
enter name: Roshani Wathore
enter age: 19
enter address: gandhi nagar
enter employee id: 676
enter Salary Factor: 55
enter hike: 858
enter technical skills: Python html css
name is: Roshani Wathore
age is::19
address is: gandhi nagar
employee id is ::676
Salary Factor is::55
hike is ::858
techincal skills is::Python
Run TODO Problems Profiler Terminal Build
Build completed successfully in 1 sec, 475 ms (4 minutes ago)
Structure
Favorites

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Algorithm :02	1. Creating the parent class employee and initialize its data members.(EmpId ,EmpSalary) and a basic function get details() to print the details. 2. Create 2 child class permanent employee and temporary employee that inherit employee class publically. 3. In this classes , create generate salary() that return the employee salary + hike in their salary 4. In main function, Create the object of derived class and print their respective details.
Program:02	

```
package com.company;
import java.util.*;

class Employee
{
 Scanner s=new Scanner(System.in);
 int emp_id;
 int emp_salary;

 Employee()
 {
  System.out.println("enter empid::");
  emp_id=s.nextInt();
  System.out.println("enter empsalary::");
  emp_salary=s.nextInt();
 }

 void getDetails()
 {
  System.out.println("EmployeeID is ::");
  System.out.println("EmployeeSalary is ::");
 }


}

class permanant_Employee extends Employee
{
 int hike;

 permanant_Employee( int increment)
 {
  hike = increment;
 }

 void getDetails()
 {
  System.out.println("EmployeeID is ::" + emp_id);
  System.out.println("Employee total salary is ::" +
generate_salary());
 }

 int generate_salary()
 {
  return (emp_salary + hike);
```

```
 }
}

class temporary_Employee extends Employee
{
 int hike;

 temporary_Employee( int increment)
 {
  hike = increment;
 }

 void getDetails()
 {
  System.out.println("EmployeeID is ::" + emp_id);
  System.out.println("Employee total salary is ::" +
generate_salary());
 }

 int generate_salary()
 {
  return (emp_salary + hike);
 }
}

public class Main {

 public static void main(String[] args) {
  permanant_Employee p = new permanant_Employee(3200);
  p.getDetails();
  temporary_Employee t = new  temporary_Employee(1600);
  t.getDetails();

 }
}
```

Output Screenshot:02