# Tutorial 9 solution

**Submitted by**

**Sabbir Hasan Rabbi**

**Id: k240175**

1. **Briefly explain the term class. In your explanation include details on the difference between an object and a class.**

A class is a blueprint or template for creating objects. It specifies a collection of actions (methods) and characteristics (variables) that the objects made from the class will possess. An Employee class, for instance, might define methods like calculatePay() and fields like name, ID, and salary in a payroll system.

On the other hand, an object is instance of a class. The object is the real thing that exists in memory, but the class specifies the behavior and structure. An object has real existence while the class is the template for that object which defines what we can do with the object. We can create as many objects as we want from a class.

1. **Consider Employee as a part of payroll system. What could be the attributes that will be needed in system?**

The attributes for the Employee class are as follows:

**String name**: The employee's name.

**int employeeID**: The unique identifier for the employee.

**String position**: The job title of the employee.

**double salary:** The employee's salary.

**String department**: The department the employee works in.

**String hireDate**: The date the employee was hired.

1. **How many constructors this class should have?**

In Java, a constructor is a unique method for initializing class objects. It can set initial values for an object's attributes and is called automatically upon object creation. This class can have 2 types of constructors.

Constructors fall into two categories:

**Default constructor:** It requires no arguments.

**Non-default or parameterized constructor:** Itrequires arguments.

1. **What are other methods this class can have?**

the methods for the employee class can be:

1. setLastName(string name)
2. setHourlyWage(num wage)
3. getLastName()
4. getHourlyWage()
5. getWeeklyPay()
6. calculateWeeklyPay()
7. **Briefly explain the purpose of the following line of code:**

**Employee employee1 = new Employee();**

Using **the default constructor**, this piece of code generates a new Employee class object with the name employee1. It initializes the object with the default settings specified in the Employee class and allocates memory for it.

1. **Create new project, package and create this class with all variables, constructors, getter, setter and other methods (to print the details of object and to calculate the pay of employee).**

package **emplyoee**;

**public** **class** emplyoee {

}

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

**private** **String** name;

**private** **int** employeeID;

**private** **String** position;

**private** **double** salary;

**private** **String** department;

**private** **String** hireDate;

**public** Employee() {

        this.name = "";

        this.employeeID = 0;

        this.position = "";

        this.salary = 0.0;

        this.department = "";

        this.hireDate = "";

    }

*// Parameterized Constructor*

**public** Employee(**String** name, **int** employeeID, **String** position, **double** salary, **String** department, **String** hireDate) {

        this.name = name;

        this.employeeID = employeeID;

        this.position = position;

        this.salary = salary;

        this.department = department;

        this.hireDate = hireDate;

    }

**public** **String** getName() {

        return name;

    }

**public** **void** setName(**String** name) {

        this.name = name;

    }

**public** **int** getEmployeeID() {

        return employeeID;

    }

**public** **void** setEmployeeID(**int** employeeID) {

        this.employeeID = employeeID;

    }

**public** **String** getPosition() {

        return position;

    }

**public** **void** setPosition(**String** position) {

        this.position = position;

    }

**public** **double** getSalary() {

        return salary;

    }

**public** **void** setSalary(**double** salary) {

        this.salary = salary;

    }

**public** **String** getDepartment() {

        return department;

    }

**public** **void** setDepartment(**String** department) {

        this.department = department;

    }

**public** **String** getHireDate() {

        return hireDate;

    }

**public** **void** setHireDate(**String** hireDate) {

        this.hireDate = hireDate;

    }

**public** **void** printDetails() {

        System.out.println("Name: " + name);

        System.out.println("Employee ID: " + employeeID);

        System.out.println("Position: " + position);

        System.out.println("Salary: " + salary);

        System.out.println("Department: " + department);

        System.out.println("Hire Date: " + hireDate);

    }

**public** **double** calculatePay(**int** hoursWorked) {

**double** payment = salary / 160 \* hoursWorked;

        return  payment ;

    }

}

1. **Create a test class which will have main method in it and will do as follows:**

* **Create five different objects of Employee class using all constructors. One should be created at least using non-parametrised constructor.**
* **For objects created with non-parametrised constructor, use setter to set the values and display the details.**
* **For all other objects, display details of all employees using other method.**

Here is the class with main method and 5 objects.

**public** **class** Test {

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

**Employee** employee1 = new Employee();

**Employee** employee2 = new Employee("John Doe", 1, "Developer", 70000, "IT", "01/01/2020");

**Employee** employee3 = new Employee("Jane Smith", 2, "Manager", 90000, "HR", "01/06/2018");

**Employee** employee4 = new Employee("Bob Brown", 3, "Analyst", 60000, "Finance", "01/03/2019");

**Employee** employee5 = new Employee("Alice White", 4, "Designer", 65000, "Design", "01/07/2021");

        employee1.setName("Charlie Green");

        employee1.setEmployeeID(5);

        employee1.setPosition("Tester");

        employee1.setSalary(55000);

        employee1.setDepartment("QA");

        employee1.setHireDate("01/02/2022");

        employee1.printDetails();

        employee2.printDetails();

        employee3.printDetails();

        employee4.printDetails();

        employee5.printDetails();

        System.out.println("Pay for " + employee1.getName() + ": " + employee1.calculatePay(160));

        System.out.println("Pay for " + employee2.getName() + ": " + employee1.calculatePay(160));

        System.out.println("Pay for " + employee3.getName() + ": " + employee1.calculatePay(160));

        System.out.println("Pay for " + employee4.getName() + ": " + employee1.calculatePay(160));

        System.out.println("Pay for " + employee5.getName() + ": " + employee1.calculatePay(160));

    }

}

1. **Rectangle Class**

|  |
| --- |
| **Vaccine** |
| **- VaccineID: int**  **- VaccineName: String**  **- manufacturer: String**  **- expiryDate: String** |
| **+ Vaccine()**  **+ Vaccine(int,String, String, String)**  **+ getVaccinationId(): int**  **+ setVaccinationId(int)**  **+ getVaccineName(): String**  **+ setVaccineName(String)**  **+ getManufacturer(): String**  **+ setManufacturer(String)**  **+ getExpiryDate(): String**  **+ setExpiryDate(String)**  **+ toString(): String** |

* 1. **How many constructors does the class Vaccine provide?**

-The class provides **two** constructors: one default constructor and one parameterized constructor.

* 1. **How many methods does the class Vaccine provide?**

It has a total of 11 methods including the constructors.

* 1. **How many instance variables does the class Vaccine have?**

The class has four instance variables.

* 1. **Briefly explain the purpose of the following two lines of code:**

**Rectangle Pfizer;**

**Pfizer = new Vaccine();**

These lines declare a reference variable Pfizer of the Vaccine class and then instantiate a new Vaccine object, assigning it to Pfizer.

* 1. **The two lines of code can be combined into one. Provide this one line of code:**

it can be combined into 1 line of code:

*Vaccine Pfizer = new Vaccine();*

* 1. **What method is used to set the VaccineID of an instance of the Rectangle class? What data type does it expect as an argument?**

The method *setVaccinationId(int vaccineID)* is used to set the Vaccine id. and it expects an int as an argument.

* 1. **What method is used to set the VaccineName of an instance of the Rectangle class? What data type does it expect as an argument?**

The method *setVaccineName(String vaccineName)*is used to set the VaccineName of an instance and it expects a String as an argument.