1. Person Class

public class Person {

private String name;

private int age;

// Constructor

public Person(String name, int age) {

this.name = name;

this.age = age;

}

// Getter methods

public String getName() {

return name;

}

public int getAge() {

return age;

}

}

2. Employee Class

public class Employee {

private int id;

private String firstName;

private String lastName;

private int salary;

// Constructor

public Employee(int id, String firstName, String lastName, int salary) {

this.id = id;

this.firstName = firstName;

this.lastName = lastName;

this.salary = salary;

}

// Getter methods

public int getID() {

return id;

}

public String getFirstName() {

return firstName;

}

public String getLastName() {

return lastName;

}

public String getName() {

return firstName + " " + lastName;

}

public int getSalary() {

return salary;

}

public int getAnnualSalary() {

return salary \* 12;

}

public void raiseSalary(int percent) {

salary += (salary \* percent) / 100;

}

@Override

public String toString() {

return "Employee[id=" + id + ", name=" + getName() + ", salary=" + salary + "]";

}

}

3. Circle Class

public class Circle { private double radius;

// No-argument constructor  
public Circle() {  
 this.radius = 1.0;  
}  
// Constructor with argument  
public Circle(double radius) {  
 this.radius = radius;  
}  
public double getCircumference() {  
 return 2 \* Math.PI \* radius;  
}  
}

4. Account Class

public class Account {

private String accountNumber;

private double balance;

// No-argument constructor

public Account() {

this.accountNumber = "0000";

this.balance = 0.0;

}

// Constructor with arguments

public Account(String accountNumber, double balance) {

this.accountNumber = accountNumber;

this.balance = balance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount);

} else {

System.out.println("Invalid deposit amount.");

}

}

public void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

System.out.println("Withdrawn: " + amount);

} else {

System.out.println("Invalid withdraw amount or insufficient balance.");

}

}

public double checkBalance() {

return balance;

}

}