1.

class University {

String universityName;

String location;

University(String universityName, String location) {

this.universityName = universityName;

this.location = location;

}

class Department {

String deptName;

Department(String deptName) {

this.deptName = deptName;

}

void showDetails() {

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

System.out.println("University: " + universityName);

System.out.println("Location: " + location);

}

}

static class ExamCell {

void conductExam(String examName) {

System.out.println("Conducting exam: " + examName);

}

void publishResults() {

System.out.println("Results published successfully");

}

}

}

public class UniversityDetails {

public static void main(String[] args) {

University uni = new University("MIT", "Cambridge");

University.Department dept = uni.new Department("Computer Science");

dept.showDetails();

System.out.println();

University.ExamCell examCell = new University.ExamCell();

examCell.conductExam("Final Semester Exam");

examCell.publishResults();

}

}

2.

import java.util.HashSet;

import java.util.Objects;

class Student {

int rollNo;

String name;

Student(int rollNo, String name) {

this.rollNo = rollNo;

this.name = name;

}

@Override

public boolean equals(Object obj) {

if (obj == null) return false;

if (obj == this) return true;

Student s = (Student) obj;

return rollNo == s.rollNo;

}

@Override

public int hashCode() {

return Objects.hash(rollNo);

}

@Override

public String toString() {

return "Student[RollNo=" + rollNo + ", Name=" + name + "]";

}

}

public class StudentHashSetDetails {

public static void main(String[] args) {

HashSet<Student> students = new HashSet<>();

Student s1 = new Student(101, "Alice");

Student s2 = new Student(102, "Bob");

Student s3 = new Student(101, "Charlie");

Student s4 = new Student(103, "Diana");

students.add(s1);

students.add(s2);

students.add(s3);

students.add(s4);

System.out.println("Total students added: 4");

System.out.println("HashSet size: " + students.size());

System.out.println();

System.out.println("Students in HashSet:");

for (Student s : students) {

System.out.println(s);

}

System.out.println();

System.out.println("s1.equals(s3): " + s1.equals(s3));

System.out.println("s1.hashCode(): " + s1.hashCode());

System.out.println("s3.hashCode(): " + s3.hashCode());

}

}

3.

class Product {

int productId;

String productName;

Product(int productId, String productName) {

this.productId = productId;

this.productName = productName;

}

@Override

public boolean equals(Object obj) {

if (obj == null) return false;

if (obj == this) return true;

Product p = (Product) obj;

return productId == p.productId;

}

@Override

public String toString() {

return "Product[ID=" + productId + ", Name=" + productName + "]";

}

}

public class ProductComparisonDemo {

public static void main(String[] args) {

Product p1 = new Product(101, "Laptop");

Product p2 = new Product(101, "Laptop");

Product p3 = p1;

Product p4 = new Product(102, "Mouse");

System.out.println("p1: " + p1);

System.out.println("p2: " + p2);

System.out.println("p1 == p2: " + (p1 == p2));

System.out.println("p1.equals(p2): " + p1.equals(p2));

System.out.println();

System.out.println("p1: " + p1);

System.out.println("p3: " + p3);

System.out.println("p1 == p3: " + (p1 == p3));

System.out.println("p1.equals(p3): " + p1.equals(p3));

System.out.println();

System.out.println("p1: " + p1);

System.out.println("p4: " + p4);

System.out.println("p1 == p4: " + (p1 == p4));

System.out.println("p1.equals(p4): " + p1.equals(p4));

}

}

4.

interface Discount {

double applyDiscount(double amount);

}

class Payment {

void processTransaction(double amount) {

class Validator {

boolean isValid(double amt) {

return amt > 0 && amt <= 100000;

}

}

Validator validator = new Validator();

if (validator.isValid(amount)) {

System.out.println("Payment amount is valid: $" + amount);

Discount discount = new Discount() {

@Override

public double applyDiscount(double amt) {

return amt \* 0.9;

}

};

double finalAmount = discount.applyDiscount(amount);

System.out.println("After 10% discount: $" + finalAmount);

System.out.println("Transaction processed successfully");

} else {

System.out.println("Invalid payment amount: $" + amount);

}

}

}

public class PaymentOut {

public static void main(String[] args) {

Payment payment = new Payment();

System.out.println("Transaction 1:");

payment.processTransaction(5000);

System.out.println();

System.out.println("Transaction 2:");

payment.processTransaction(-100);

System.out.println();

System.out.println("Transaction 3:");

payment.processTransaction(150000);

}

}

5.

import java.util.ArrayList;

class Book {

String title;

String author;

Book(String title, String author) {

this.title = title;

this.author = author;

}

@Override

public String toString() {

return title + " by " + author;

}

}

class Library implements Cloneable {

ArrayList<Book> books;

Library() {

books = new ArrayList<>();

}

void addBook(Book book) {

books.add(book);

}

@Override

public Library clone() throws CloneNotSupportedException {

return (Library) super.clone();

}

public Library deepClone() {

Library newLibrary = new Library();

for (Book book : this.books) {

newLibrary.addBook(new Book(book.title, book.author));

}

return newLibrary;

}

void displayBooks() {

for (Book book : books) {

System.out.println(book);

}

}

}

public class LibraryCloning {

public static void main(String[] args) throws CloneNotSupportedException {

Library lib1 = new Library();

lib1.addBook(new Book("Java Basics", "John"));

lib1.addBook(new Book("Python Guide", "Sarah"));

Library lib2 = lib1.clone();

Library lib3 = lib1.deepClone();

System.out.println("Original Library:");

lib1.displayBooks();

System.out.println();

lib1.books.get(0).title = "Advanced Java";

System.out.println("After modifying original:");

System.out.println("Original Library:");

lib1.displayBooks();

System.out.println();

System.out.println("Shallow Copy Library:");

lib2.displayBooks();

System.out.println();

System.out.println("Deep Copy Library:");

lib3.displayBooks();

}

}

6.

class Employee {

int id;

String name;

double salary;

Employee(int id, String name, double salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

@Override

public String toString() {

return "Employee[ID=" + id + ", Name=" + name + ", Salary=$" + salary + "]";

}

}

public class EmployeeDetails {

public static void main(String[] args) {

Employee emp1 = new Employee(101, "John", 50000);

Employee emp2 = new Employee(102, "Sarah", 60000);

Employee emp3 = new Employee(103, "Mike", 55000);

System.out.println(emp1);

System.out.println("Class name: " + emp1.getClass().getName());

System.out.println();

System.out.println(emp2);

System.out.println("Class name: " + emp2.getClass().getName());

System.out.println();

System.out.println(emp3);

System.out.println("Class name: " + emp3.getClass().getName());

}

}