# **Best Programming Practice**

1. Use **static** for shared values and utility methods to reduce memory usage and avoid redundancy.
2. Leverage **this** to avoid ambiguity when initializing attributes.
3. Declare **final** variables for identifiers or constants that should remain unchanged.
4. Use **instanceof** for safe type-checking and to prevent runtime errors during typecasting.

### **Sample Program 1: Bank Account System**

Create a BankAccount class with the following features:

1. **Static**:

○ A static variable bankName is shared across all accounts.

○ A static method getTotalAccounts() to display the total number of accounts.

1. **This**:

○ Use this to resolve ambiguity in the constructor when initializing accountHolderName and accountNumber.

1. **Final**:

○ Use a final variable accountNumber to ensure it cannot be changed once assigned.

1. **Instanceof**:

○ Check if an account object is an instance of the BankAccount class before displaying its details.

**package com.bridgelabz.oops.inbuiltfunction;**

**public class BankAccount {**

**// Static variable shared across all accounts**

**private static String *bankName* = "State bank of India";**

**private static int *totalAccounts* = 0;**

**// Final variable to ensure account number cannot be changed once assigned**

**private final String accountNumber;**

**private String accountHolderName;**

**private double balance;**

**// Constructor using 'this' to avoid ambiguity**

**public BankAccount(String accountHolderName, String accountNumber, double balance) {**

**this.accountHolderName = accountHolderName;**

**this.accountNumber = accountNumber; // Final variable assigned only once**

**this.balance = balance;**

***totalAccounts*++; // Increment total accounts with each new account**

**}**

**// Static method to get the total number of accounts**

**public static void getTotalAccounts() {**

**System.*out*.println("Total number of accounts: " + *totalAccounts*);**

**}**

**// Method to display account details**

**public void displayAccountDetails() {**

**// Using instanceof to check if the object is a BankAccount instance**

**if (this instanceof BankAccount) {**

**System.*out*.println("Bank Name: " + *bankName*);**

**System.*out*.println("Account Holder: " + accountHolderName);**

**System.*out*.println("Account Number: " + accountNumber);**

**System.*out*.println("Balance: $" + balance);**

**} else {**

**System.*out*.println("Invalid account instance.");**

**}**

**}**

**// Getters and setters**

**public String getAccountHolderName() {**

**return accountHolderName;**

**}**

**public void setAccountHolderName(String accountHolderName) {**

**this.accountHolderName = accountHolderName;**

**}**

**public double getBalance() {**

**return 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.*err*.println("Insufficient balance or invalid amount.");**

**}**

**}**

**// Main method for testing**

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

**// Create two bank accounts**

**BankAccount account1 = new BankAccount("Lynda Princy", "ACC12345", 1000.00);**

**BankAccount account2 = new BankAccount("Prince Danish", "ACC67890", 500.00);**

**// Display account details**

**System.*out*.println("\nAccount 1 Details:");**

**account1.displayAccountDetails();**

**System.*out*.println("\nAccount 2 Details:");**

**account2.displayAccountDetails();**

**// Check total accounts**

**BankAccount.*getTotalAccounts*();**

**// Deposit and withdraw operations**

**System.*out*.println("\nPerforming transactions on Account 1:");**

**account1.deposit(200);**

**account1.withdraw(150);**

**account1.displayAccountDetails();**

**account2.displayAccountDetails();**

**account2.deposit(100);**

**account2.withdraw(800);**

**}**

**}**

**o/p:**

**Account 1 Details:**

**Bank Name: State bank of India**

**Account Holder: Lynda Princy**

**Account Number: ACC12345**

**Balance: $1000.0**

**Account 2 Details:**

**Bank Name: State bank of India**

**Account Holder: Prince Danish**

**Account Number: ACC67890**

**Balance: $500.0**

**Total number of accounts: 2**

**Performing transactions on Account 1:**

**Deposited: $200.0**

**Withdrawn: $150.0**

**Bank Name: State bank of India**

**Account Holder: Lynda Princy**

**Account Number: ACC12345**

**Balance: $1050.0**

**Bank Name: State bank of India**

**Account Holder: Prince Danish**

**Account Number: ACC67890**

**Balance: $500.0**

**Deposited: $100.0**

**Insufficient balance or invalid amount.**

### **Sample Program 2: Library Management System**

Create a Book class to manage library books with the following features:

1. **Static**:

○ A static variable libraryName shared across all books.

○ A static method displayLibraryName() to print the library name.

1. **This**:

○ Use this to initialize title, author, and isbn in the constructor.

1. **Final**:

○ Use a final variable isbn to ensure the unique identifier of a book cannot be changed.

1. **Instanceof**:

○ Verify if an object is an instance of the Book class before displaying its details.

**package com.bridgelabz.oops.inbuiltfunction;**

**class Book {**

**// Static variable shared across all instances**

**private static String *libraryName*;**

**// Instance variables**

**private String title;**

**private String author;**

**private final String isbn; // Final variable to ensure immutability**

**// Constructor to initialize instance variables**

**public Book(String title, String author, String isbn) {**

**this.title = title;**

**this.author = author;**

**this.isbn = isbn;**

**}**

**// Static method to set the library name**

**public static void setLibraryName(String name) {**

***libraryName* = name;**

**}**

**// Static method to display the library name**

**public static void displayLibraryName() {**

**System.*out*.println("Library Name: " + *libraryName*);**

**}**

**// Method to display book details**

**public void displayBookDetails() {**

**// Using 'instanceof' to verify the object's type before displaying details**

**if (this instanceof Book) {**

**System.*out*.println("Title: " + title);**

**System.*out*.println("Author: " + author);**

**System.*out*.println("ISBN: " + isbn);**

**} else {**

**System.*out*.println("The object is not an instance of the Book class.");**

**}**

**}**

**// Getters for instance variables (optional, but recommended for encapsulation)**

**public String getTitle() {**

**return title;**

**}**

**public String getAuthor() {**

**return author;**

**}**

**public String getIsbn() {**

**return isbn;**

**}**

**}**

**public class Library {**

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

**// Set the library name (static variable)**

**Book.*setLibraryName*("Egmore Library");**

**// Display the library name**

**Book.*displayLibraryName*();**

**// Create a new book instance**

**Book book1 = new Book("Effective Java", "Joshua Bloch", "978-0134685991");**

**// Display book details**

**book1.displayBookDetails();**

**}**

**}**

**0/p:**

**Library Name: Egmore Library**

**Title: Effective Java**

**Author: Joshua Bloch**

**ISBN: 978-0134685991**

### 

### **Sample Program 3: Employee Management System**

Design an Employee class with the following features:

1. **Static**:

○ A static variable companyName shared by all employees.

○ A static method displayTotalEmployees() to show the total number of employees.

1. **This**:

○ Use this to initialize name, id, and designation in the constructor.

1. **Final**:

○ Use a final variable id for the employee ID, which cannot be modified after assignment.

1. **Instanceof**

○ Check if a given object is an instance of the Employee class before printing the employee details.

**package com.bridgelabz.oops.inbuiltfunction;**

**public class Company {**

**// Static variable shared by all employees**

**private static String *companyName*;**

**private static int *totalEmployees* = 0;**

**// Instance variables**

**private String name;**

**private final int id; // Final variable to ensure the employee ID cannot be modified**

**private String designation;**

**// Constructor to initialize name, id, and designation**

**public Company(String name, int id, String designation) {**

**this.name = name;**

**this.id = id;**

**this.designation = designation;**

***totalEmployees*++;**

**}**

**// Static method to set the company name**

**public static void setCompanyName(String name) {**

***companyName* = name;**

**}**

**// Static method to display the total number of employees**

**public static void displayTotalEmployees() {**

**System.*out*.println("Total Employees: " + *totalEmployees*);**

**}**

**// Method to display employee details**

**public void displayEmployeeDetails() {**

**// Using 'instanceof' to verify the object's type before displaying details**

**if (this instanceof Company) {**

**System.*out*.println("Company Name: " + *companyName*);**

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

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

**System.*out*.println("Designation: " + designation);**

**} else {**

**System.*out*.println("The object is not an instance of the Employee class.");**

**}**

**}**

**// Getters for instance variables (optional, but recommended for encapsulation)**

**public String getName() {**

**return name;**

**}**

**public int getId() {**

**return id;**

**}**

**public String getDesignation() {**

**return designation;**

**}**

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

**// Set the company name (static variable)**

**Company.*setCompanyName*("Tech Solutions Inc.");**

**// Create new employee instances**

**Company emp1 = new Company("Thamarai", 101, "Software Engineer");**

**Company emp2 = new Company("Rohan", 102, "Project Manager");**

**// Display total number of employees**

**Company.*displayTotalEmployees*();**

**// Display employee details**

**emp1.displayEmployeeDetails();**

**emp2.displayEmployeeDetails();**

**}**

**}**

**o/p:**

**Total Employees: 2**

**Company Name: Tech Solutions Inc.**

**Employee ID: 101**

**Name: Thamarai**

**Designation: Software Engineer**

**Company Name: Tech Solutions Inc.**

**Employee ID: 102**

**Name: Rohan**

**Designation: Project Manager**

### **Sample Program 4: Shopping Cart System**

Create a Product class to manage shopping cart items with the following features:

1. **Static**:

○ A static variable discount shared by all products.

○ A static method updateDiscount() to modify the discount percentage.

1. **This**:

○ Use this to initialize productName, price, and quantity in the constructor.

1. **Final**:

○ Use a final variable productID to ensure each product has a unique identifier that cannot be changed.

1. **Instanceof**:

○ Validate whether an object is an instance of the Product class before processing its details.

**package com.bridgelabz.oops.inbuiltfunction;**

**public class Product {**

**// Static variable shared by all products**

**private static double *discount* = 0.0;**

**// Instance variables**

**private final String productID; // Final variable to ensure the unique identifier cannot be changed**

**private String productName;**

**private double price;**

**private int quantity;**

**// Constructor to initialize productName, price, and quantity**

**public Product(String productID, String productName, double price, int quantity) {**

**this.productID = productID;**

**this.productName = productName;**

**this.price = price;**

**this.quantity = quantity;**

**}**

**// Static method to update the discount percentage**

**public static void updateDiscount(double newDiscount) {**

**if (newDiscount >= 0.0 && newDiscount <= 100.0) {**

***discount* = newDiscount;**

**} else {**

**System.*out*.println("Invalid discount percentage. Please enter a value between 0 and 100.");**

**}**

**}**

**// Method to calculate the price after applying the discount**

**public double getDiscountedPrice() {**

**return price - (price \* *discount* / 100);**

**}**

**// Method to display product details**

**public void displayProductDetails() {**

**// Using 'instanceof' to verify the object's type before displaying details**

**if (this instanceof Product) {**

**System.*out*.println("Product ID: " + productID);**

**System.*out*.println("Product Name: " + productName);**

**System.*out*.println("Price: $" + price);**

**System.*out*.println("Quantity: " + quantity);**

**System.*out*.println("Discount: " + *discount* + "%");**

**System.*out*.println("Price after Discount: $" + getDiscountedPrice());**

**} else {**

**System.*out*.println("The object is not an instance of the Product class.");**

**}**

**}**

**// Getters and setters for instance variables (optional, but recommended for encapsulation)**

**public String getProductID() {**

**return productID;**

**}**

**public String getProductName() {**

**return productName;**

**}**

**public void setProductName(String productName) {**

**this.productName = productName;**

**}**

**public double getPrice() {**

**return price;**

**}**

**public void setPrice(double price) {**

**this.price = price;**

**}**

**public int getQuantity() {**

**return quantity;**

**}**

**public void setQuantity(int quantity) {**

**this.quantity = quantity;**

**}**

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

**// Update the discount for all products**

**Product.*updateDiscount*(10.0); // 10% discount**

**// Create new product instances**

**Product product1 = new Product("P001", "Laptop", 1200.00, 5);**

**Product product2 = new Product("P002", "Smartphone", 800.00, 10);**

**// Display product details**

**product1.displayProductDetails();**

**System.*out*.println();**

**product2.displayProductDetails();**

**}**

**}**

**o/p:**

**Product ID: P001**

**Product Name: Laptop**

**Price: $1200.0**

**Quantity: 5**

**Discount: 10.0%**

**Price after Discount: $1080.0**

**Product ID: P002**

**Product Name: Smartphone**

**Price: $800.0**

**Quantity: 10**

**Discount: 10.0%**

**Price after Discount: $720.0**

### **Sample Program 5: University Student Management**

Create a Student class to manage student data with the following features:

1. **Static**:

○ A static variable universityName shared across all students.

○ A static method displayTotalStudents() to show the number of students enrolled.

1. **This**:

○ Use this in the constructor to initialize name, rollNumber, and grade.

1. **Final**:

○ Use a final variable rollNumber for each student that cannot be changed.

1. **Instanceof**:

○ Check if a given object is an instance of the Student class before performing operations like displaying or updating grades.

**package com.bridgelabz.oops.inbuiltfunction;**

**public class Student {**

**// Static variable shared across all instances**

**private static String *universityName*;**

**private static int *totalStudents* = 0;**

**// Instance variables**

**private final int rollNumber; // Final variable to ensure the roll number cannot be changed**

**private String name;**

**private char grade;**

**// Constructor to initialize name, rollNumber, and grade**

**public Student(String name, int rollNumber, char grade) {**

**this.name = name;**

**this.rollNumber = rollNumber;**

**this.grade = grade;**

***totalStudents*++;**

**}**

**// Static method to set the university name**

**public static void setUniversityName(String name) {**

***universityName* = name;**

**}**

**// Static method to display the total number of students enrolled**

**public static void displayTotalStudents() {**

**System.*out*.println("Total Students Enrolled: " + *totalStudents*);**

**}**

**// Method to display student details**

**public void displayStudentDetails() {**

**// Using 'instanceof' to verify the object's type before displaying details**

**if (this instanceof Student) {**

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

**System.*out*.println("Roll Number: " + rollNumber);**

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

**System.*out*.println("Grade: " + grade);**

**} else {**

**System.*out*.println("The object is not an instance of the Student class.");**

**}**

**}**

**// Method to update the student's grade**

**public void updateGrade(char newGrade) {**

**// Using 'instanceof' to verify the object's type before updating grade**

**if (this instanceof Student) {**

**this.grade = newGrade;**

**System.*out*.println("Grade updated to: " + newGrade);**

**} else {**

**System.*out*.println("The object is not an instance of the Student class.");**

**}**

**}**

**// Getters for instance variables (optional, but recommended for encapsulation)**

**public String getName() {**

**return name;**

**}**

**public int getRollNumber() {**

**return rollNumber;**

**}**

**public char getGrade() {**

**return grade;**

**}**

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

**// Set the university name (static variable)**

**Student.*setUniversityName*("Global University");**

**// Create new student instances**

**Student student1 = new Student("Hemashree", 101, 'A');**

**Student student2 = new Student("Sharmila", 102, 'B');**

**// Display total number of students enrolled**

**Student.*displayTotalStudents*();**

**// Display student details**

**student1.displayStudentDetails();**

**System.*out*.println();**

**student2.displayStudentDetails();**

**// Update student grade**

**student2.updateGrade('A');**

**System.*out*.println();**

**// Display updated student details**

**student2.displayStudentDetails();**

**}**

**}**

**o/p:**

**Total Students Enrolled: 2**

**University Name: Global University**

**Roll Number: 101**

**Name: Hemashree**

**Grade: A**

**University Name: Global University**

**Roll Number: 102**

**Name: Sharmila**

**Grade: B**

**Grade updated to: A**

**University Name: Global University**

**Roll Number: 102**

**Name: Sharmila**

**Grade: A**

### **Sample Program 6: Vehicle Registration System**

Create a Vehicle class with the following features:

1. **Static**:

○ A static variable registrationFee common for all vehicles.

○ A static method updateRegistrationFee() to modify the fee.

1. **This**:

○ Use this to initialize ownerName, vehicleType, and registrationNumber in the constructor.

1. **Final**:

○ Use a final variable registrationNumber to uniquely identify each vehicle.

1. **Instanceof**:

○ Check if an object belongs to the Vehicle class before displaying its registration

○ details.

**package com.bridgelabz.oops.inbuiltfunction;**

**public class Vehicle {**

**// Static variable shared by all vehicles**

**private static double *registrationFee* = 100.0; // Default registration fee**

**// Instance variables**

**private final String registrationNumber; // Final variable to ensure the registration number cannot be changed**

**private String ownerName;**

**private String vehicleType;**

**// Constructor to initialize ownerName, vehicleType, and registrationNumber**

**public Vehicle(String ownerName, String vehicleType, String registrationNumber) {**

**this.ownerName = ownerName;**

**this.vehicleType = vehicleType;**

**this.registrationNumber = registrationNumber;**

**}**

**// Static method to update the registration fee**

**public static void updateRegistrationFee(double newFee) {**

**if (newFee >= 0.0) {**

***registrationFee* = newFee;**

**} else {**

**System.*out*.println("Invalid registration fee. Please enter a non-negative value.");**

**}**

**}**

**// Method to display vehicle registration details**

**public void displayRegistrationDetails() {**

**// Using 'instanceof' to verify the object's type before displaying details**

**if (this instanceof Vehicle) {**

**System.*out*.println("Owner Name: " + ownerName);**

**System.*out*.println("Vehicle Type: " + vehicleType);**

**System.*out*.println("Registration Number: " + registrationNumber);**

**System.*out*.println("Registration Fee: $" + *registrationFee*);**

**} else {**

**System.*out*.println("The object is not an instance of the Vehicle class.");**

**}**

**}**

**// Getters for instance variables (optional, but recommended for encapsulation)**

**public String getOwnerName() {**

**return ownerName;**

**}**

**public String getVehicleType() {**

**return vehicleType;**

**}**

**public String getRegistrationNumber() {**

**return registrationNumber;**

**}**

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

**// Update the registration fee for all vehicles**

**Vehicle.*updateRegistrationFee*(150.0); // Set registration fee to $150.0**

**// Create new vehicle instances**

**Vehicle vehicle1 = new Vehicle("Honest raj", "Sedan", "ABC123");**

**Vehicle vehicle2 = new Vehicle("Price danish", "SUV", "XYZ789");**

**// Display vehicle registration details**

**vehicle1.displayRegistrationDetails();**

**System.*out*.println();**

**vehicle2.displayRegistrationDetails();**

**}**

**}**

**o/p:**

**Owner Name: Honest raj**

**Vehicle Type: Sedan**

**Registration Number: ABC123**

**Registration Fee: $150.0**

**Owner Name: Price danish**

**Vehicle Type: SUV**

**Registration Number: XYZ789**

**Registration Fee: $150.0**

### **Sample Program 7: Hospital Management System**

Create a Patient class with the following features:

1. **Static**:

○ A static variable hospitalName shared among all patients.

○ A static method getTotalPatients() to count the total patients admitted.

1. **This**:

○ Use this to initialize name, age, and ailment in the constructor.

1. **Final**:

○ Use a final variable patientID to uniquely identify each patient.

1. **Instanceof**:

○ Check if an object is an instance of the Patient class before displaying its details.

**package com.bridgelabz.oops.inbuiltfunction;**

**public class Patient {**

**// Static variable shared among all patients**

**private static String *hospitalName*;**

**private static int *totalPatients* = 0;**

**// Instance variables**

**private final String patientID; // Final variable to uniquely identify each patient**

**private String name;**

**private int age;**

**private String ailment;**

**// Constructor to initialize name, age, ailment, and patientID**

**public Patient(String name, int age, String ailment, String patientID) {**

**this.name = name;**

**this.age = age;**

**this.ailment = ailment;**

**this.patientID = patientID;**

***totalPatients*++;**

**}**

**// Static method to set the hospital name**

**public static void setHospitalName(String name) {**

***hospitalName* = name;**

**}**

**// Static method to get the total number of patients admitted**

**public static int getTotalPatients() {**

**return *totalPatients*;**

**}**

**// Method to display patient details**

**public void displayPatientDetails() {**

**// Using 'instanceof' to verify the object's type before displaying details**

**if (this instanceof Patient) {**

**System.*out*.println("Hospital Name: " + *hospitalName*);**

**System.*out*.println("Patient ID: " + patientID);**

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

**System.*out*.println("Age: " + age);**

**System.*out*.println("Ailment: " + ailment);**

**} else {**

**System.*out*.println("The object is not an instance of the Patient class.");**

**}**

**}**

**// Getters for instance variables (optional, but recommended for encapsulation)**

**public String getName() {**

**return name;**

**}**

**public int getAge() {**

**return age;**

**}**

**public String getAilment() {**

**return ailment;**

**}**

**public String getPatientID() {**

**return patientID;**

**}**

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

**// Set the hospital name (static variable)**

**Patient.*setHospitalName*("City Hospital");**

**// Create new patient instances**

**Patient patient1 = new Patient("Lathika", 30, "Flu", "P001");**

**Patient patient2 = new Patient("Lidiya", 45, "Fracture", "P002");**

**// Display total number of patients admitted**

**System.*out*.println("Total Patients Admitted: " + Patient.*getTotalPatients*());**

**// Display patient details**

**patient1.displayPatientDetails();**

**System.*out*.println();**

**patient2.displayPatientDetails();**

**}**

**}**

**o/p:**

**Total Patients Admitted: 2**

**Hospital Name: City Hospital**

**Patient ID: P001**

**Name: Lathika**

**Age: 30**

**Ailment: Flu**

**Hospital Name: City Hospital**

**Patient ID: P002**

**Name: Lidiya**

**Age: 45**

**Ailment: Fracture**