

# Java OOPs – Object, Class and Constructor Assignments

## Assignment 1: Student Class

```
package oops;

public class Student {
    int id;
    String name;

    void displayDetails() {
        System.out.println(id + " " + name);
    }

    public static void main(String[] args) {
        Student s = new Student();
        s.id = 101;
        s.name = "Rahul";
        s.displayDetails();
    }
}
```

### Output:

101 Rahul

## Assignment 2: Employee Salary (Default Constructor)

```
package oops;

public class Employee {
    int empId;
    String empName;
    double salary;

    Employee() {
        empId = 1;
        empName = "Anita";
        salary = 30000;
    }

    void display() {
        System.out.println(empId + " " + empName + " " + salary);
    }

    public static void main(String[] args) {
        Employee e = new Employee();
        e.display();
    }
}
```

### Output:

1 Anita 30000.0

## Assignment 3: Car Information (Parameterized Constructor)

```
package oops;

public class Car {
    String brand, model;
```

```

double price;

Car(String b, String m, double p) {
    brand = b;
    model = m;
    price = p;
}

void display() {
    System.out.println(brand + " " + model + " " + price);
}

public static void main(String[] args) {
    Car c1 = new Car("Honda", "City", 1200000);
    Car c2 = new Car("Hyundai", "i20", 900000);
    c1.display();
    c2.display();
}
}

```

**Output:**

```

Honda City 1200000.0
Hyundai i20 900000.0

```

## Assignment 4: Book Details (Constructor Overloading)

```

package oops;

public class Book {
    int bookId;
    String title;

    Book() {
        bookId = 1;
        title = "Java Basics";
    }

    Book(int id, String t) {
        bookId = id;
        title = t;
    }

    void display() {
        System.out.println(bookId + " " + title);
    }

    public static void main(String[] args) {
        Book b1 = new Book();
        Book b2 = new Book(2, "OOPs Java");
        b1.display();
        b2.display();
    }
}

```

**Output:**

```

1 Java Basics
2 OOPs Java

```

## Assignment 5: Bank Account (Object Initialization)

```

package oops;

```

```

public class BankAccount {
    int accountNumber;
    String holderName;
    double balance;

    BankAccount(int a, String h, double b) {
        accountNumber = a;
        holderName = h;
        balance = b;
    }

    void showAccount() {
        System.out.println(accountNumber + " " + holderName + " " + balance);
    }

    public static void main(String[] args) {
        BankAccount acc = new BankAccount(123, "Ravi", 5000);
        acc.showAccount();
    }
}

```

**Output:**

123 Ravi 5000.0

## Assignment 6: Product Management (this keyword)

```

package oops;

public class Product {
    int productId;
    String productName;
    double price;

    Product(int productId, String productName, double price) {
        this.productId = productId;
        this.productName = productName;
        this.price = price;
    }

    void display() {
        System.out.println(productId + " " + productName + " " + price);
    }

    public static void main(String[] args) {
        Product p = new Product(1, "Laptop", 60000);
        p.display();
    }
}

```

**Output:**

1 Laptop 60000.0

## Assignment 7: User Login System (Constructor Overloading)

```

package oops;

public class User {
    String email, password;

    User(String email) {
        this.email = email;
    }
}

```

```

        this.password = "default";
    }

    User(String email, String password) {
        this.email = email;
        this.password = password;
    }

    void display() {
        System.out.println(email + " " + password);
    }

    public static void main(String[] args) {
        User u1 = new User("user@mail.com");
        User u2 = new User("admin@mail.com", "admin123");
        u1.display();
        u2.display();
    }
}

```

### Output:

```

user@mail.com default
admin@mail.com admin123

```

## Assignment 8: Mobile Store

```

package oops;

public class Mobile {
    String brand;
    int ram, storage;
    double price;

    Mobile(String b, int r) {
        brand = b;
        ram = r;
        storage = 64;
        price = 15000;
    }

    Mobile(String b, int r, int s, double p) {
        brand = b;
        ram = r;
        storage = s;
        price = p;
    }

    void display() {
        System.out.println(brand + " " + ram + "GB " + storage + "GB " + price);
    }

    public static void main(String[] args) {
        Mobile m1 = new Mobile("Samsung", 4);
        Mobile m2 = new Mobile("Apple", 8, 128, 80000);
        m1.display();
        m2.display();
    }
}

```

### Output:

```

Samsung 4GB 64GB 15000.0
Apple 8GB 128GB 80000.0

```

## Assignment 9: Library Book System

```
package oops;

public class LibraryBook {
    int bookId;
    String title, author;

    LibraryBook(int id, String t, String a) {
        bookId = id;
        title = t;
        author = a;
    }

    boolean isAvailable() {
        return true;
    }

    public static void main(String[] args) {
        LibraryBook b = new LibraryBook(1, "Java", "James");
        System.out.println(b.isAvailable());
    }
}
```

### Output:

true

## Assignment 10: College Admission

```
package oops;

public class StudentAdmission {
    String type;

    StudentAdmission() {
        type = "General Admission";
    }

    StudentAdmission(int marks) {
        type = marks >= 80 ? "Merit Admission" : "General Admission";
    }

    void display() {
        System.out.println(type);
    }

    public static void main(String[] args) {
        StudentAdmission s1 = new StudentAdmission();
        StudentAdmission s2 = new StudentAdmission(85);
        s1.display();
        s2.display();
    }
}
```

### Output:

General Admission  
Merit Admission

## Assignment 11: Constructor Chaining (this)

```
package oops;

public class Person {
    String name;
    int age;

    Person(String name) {
        this(name, 18);
    }

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public static void main(String[] args) {
        Person p = new Person("Rahul");
        System.out.println(p.name + " " + p.age);
    }
}
```

**Output:**

Rahul 18

## Assignment 12: E-Commerce Order

```
package oops;

public class Order {
    int orderId;
    String customerName;
    double amount;

    Order(int id, String name, double amt) {
        orderId = id;
        customerName = name;
        amount = amt + (amt * 0.18);
    }

    void display() {
        System.out.println(orderId + " " + customerName + " " + amount);
    }

    public static void main(String[] args) {
        Order o = new Order(1, "Amit", 1000);
        o.display();
    }
}
```

**Output:**

1 Amit 1180.0