

Assignment 1: Calculator Overloading

(Compile-time polymorphism)

```
class Calculator {  
  
    int add(int a, int b) {  
        return a + b;  
    }  
  
    int add(int a, int b, int c) {  
        return a + b + c;  
    }  
  
    double add(double a, double b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {  
        Calculator c = new Calculator();  
        System.out.println(c.add(10, 20));  
        System.out.println(c.add(10, 20, 30));  
        System.out.println(c.add(5.5, 4.5));  
    }  
}
```

Assignment 2: Area Calculator

(Method signature change)

```
class Area {
```

```
    void area(int side) {
```

```

        System.out.println("Square area: " + (side * side));
    }

void area(int length, int breadth) {
    System.out.println("Rectangle area: " + (length * breadth));
}

void area(double radius) {
    System.out.println("Circle area: " + (3.14 * radius * radius));
}

public static void main(String[] args) {
    Area a = new Area();
    a.area(5);
    a.area(10, 20);
    a.area(7.0);
}
}

```

Assignment 3: Print Data

(Overloaded methods)

```

class Printer {

    void print(int a) {
        System.out.println(a);
    }

    void print(String s) {
        System.out.println(s);
    }
}

```

```
void print(int a, String s) {  
    System.out.println(a + " " + s);  
}  
  
public static void main(String[] args) {  
    Printer p = new Printer();  
    p.print(10);  
    p.print("Hello");  
    p.print(5, "Java");  
}  
}
```

Assignment 4: Login System

(Overloading for flexibility)

```
class Login {  
  
    void login(String email) {  
        System.out.println("Login with email: " + email);  
    }  
  
    void login(String email, String password) {  
        System.out.println("Login with email & password");  
    }  
  
    public static void main(String[] args) {  
        Login l = new Login();  
        l.login("abc@gmail.com");  
        l.login("abc@gmail.com", "12345");  
    }  
}
```

```
}
```

Assignment 5: Payment Calculation

(Same method, different parameters)

```
class Payment {
```

```
    void pay(int amount) {
```

```
        System.out.println("Paid amount: " + amount);
```

```
}
```

```
    void pay(int amount, String mode) {
```

```
        System.out.println("Paid " + amount + " via " + mode);
```

```
}
```

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

```
        Payment p = new Payment();
```

```
        p.pay(500);
```

```
        p.pay(1000, "UPI");
```

```
}
```

```
}
```

Assignment 6: Shape Drawing

(Runtime polymorphism)

```
class Shape {
```

```
    void draw() {
```

```
        System.out.println("Drawing shape");
```

```
}
```

```
}
```

```
class Circle extends Shape {
```

```
    void draw() {
```

```

        System.out.println("Drawing circle");
    }

}

class Rectangle extends Shape {
    void draw() {
        System.out.println("Drawing rectangle");
    }
}

class Main {
    public static void main(String[] args) {
        Shape s;
        s = new Circle();
        s.draw();

        s = new Rectangle();
        s.draw();
    }
}

```

Assignment 7: Bank Interest

(Dynamic method dispatch)

```

class Bank {
    int getInterestRate() {
        return 5;
    }
}

```

```

class SBI extends Bank {
    int getInterestRate() {
        return 6;
    }
}

class HDFC extends Bank {
    int getInterestRate() {
        return 7;
    }
}

class Main {
    public static void main(String[] args) {
        Bank b;

        b = new SBI();
        System.out.println("SBI Interest: " + b.getInterestRate());

        b = new HDFC();
        System.out.println("HDFC Interest: " + b.getInterestRate());
    }
}

```

**Assignment 8: Notification System
(Real-time example – runtime polymorphism)**

```

class Notification {
    void send() {
        System.out.println("Sending notification");
    }
}

```

```
}
```

```
class EmailNotification extends Notification {  
    void send() {  
        System.out.println("Sending Email Notification");  
    }  
}
```

```
class SMSNotification extends Notification {  
    void send() {  
        System.out.println("Sending SMS Notification");  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Notification n;  
  
        n = new EmailNotification();  
        n.send();  
  
        n = new SMSNotification();  
        n.send();  
    }  
}
```