

Polymorphism

Direct Challenges

1. Demonstrate method overloading with different parameters

```
public class Calculator {  
    int add(int a, int b) {  
        return a + b;  
    }  
  
    double add(double a, double b) {  
        return a + b;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Calculator calc = new Calculator();  
        System.out.println("Sum (int): " + calc.add(5, 3));  
        System.out.println("Sum (double): " + calc.add(2.5, 3.7));  
    }  
}
```

Output:

Sum (int): 8

Sum (double): 6.2

2. Show method overriding in child class

```
class Animal {  
    void sound() {
```

```
        System.out.println("Animal makes a sound");
    }
}
```

```
class Cat extends Animal {
    @Override
    void sound() {
        System.out.println("Cat meows");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Animal a = new Cat();
        a.sound();
    }
}
```

Output:

Cat meows

3. Use instanceof to check the type at runtime

```
class Animal {}
```

```
class Dog extends Animal {}
```

```
public class Main {
    public static void main(String[] args) {
        Animal a = new Dog();
        System.out.println(a instanceof Dog); // true
    }
}
```

```
        System.out.println(a instanceof Animal); // true
    }
}
```

Output:

true

true

Scenario-Based Challenges

1. Create a Payment class and override pay() in CreditCard, Cash subclasses

```
class Payment {
    void pay() {
        System.out.println("Generic payment processing");
    }
}

class CreditCard extends Payment {
    @Override
    void pay() {
        System.out.println("Payment done using Credit Card");
    }
}

class Cash extends Payment {
    @Override
    void pay() {
        System.out.println("Payment done using Cash");
    }
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Payment p1 = new CreditCard();  
        Payment p2 = new Cash();  
        p1.pay();  
        p2.pay();  
    }  
}
```

Output:

Payment done using Credit Card

Payment done using Cash

2. Build a Notification class and implement different behaviors for Email, SMS, Push

```
class Notification {  
    void notifyUser() {  
        System.out.println("Sending a notification");  
    }  
}  
  
class Email extends Notification {  
    @Override  
    void notifyUser() {  
        System.out.println("Sending Email Notification");  
    }  
}  
  
class SMS extends Notification {  
    @Override  
    void notifyUser() {
```

```
        System.out.println("Sending SMS Notification");
    }
}
```

```
class Push extends Notification {
    @Override
    void notifyUser() {
        System.out.println("Sending Push Notification");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Notification n1 = new Email();
        Notification n2 = new SMS();
        Notification n3 = new Push();
        n1.notifyUser();
        n2.notifyUser();
        n3.notifyUser();
    }
}
```

Output:

Sending Email Notification

Sending SMS Notification

Sending Push Notification