```
        this.accountNumber = accountNumber;
        this.accountHolderName = accountHolderName;
        this.balance = initialDeposit;
    }
    public void displayInfo() {
        System.out.println(accountHolderName + " (Account: " + accountNumber + ")");
        System.out.printf("Balance: $%.2f\n", balance);
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.printf("Transaction successful. New balance: $%.2f\n", balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
    public void withdraw(double amount) {
        System.out.println("Withdrawal not supported in base account.");
    }
}
class SavingsAccount extends Account {
    private double interestRate;
    public SavingsAccount(String accountNumber, String accountHolderName, double initialDeposi
        super(accountNumber, accountHolderName, initialDeposit);
        this.interestRate = interestRate;
    }
    @Override
    public void displayInfo() {
        super.displayInfo();
```

```
Transaction successful. New balance: $-1000.00
(overdraft limit reached)

Final account statuses:
Jane Doe (Account: 123456789)
Balance: $7500.00
Interest Rate: 1.5%
Type: Savings Account

John Smith (Account: 987654321)
Balance: $-1000.00
Overdraft Limit: $1000.00
Type: Checking Account
```

```java
        }
        return sum;
    }

    public static double sum(double... numbers) {
        double sum = 0.0;
        for (double num : numbers) {
            sum += num;
        }
        return sum;
    }

    public static void main(String[] args) {
        int intSum = sum(5,1,2,4,8);
        System.out.println("Sum of integers: " + intSum);

        double doubleSum = sum(1.5,1.5, 2.4);
        System.out.println("Sum of doubles: " + doubleSum);
    }
}
```

```
Sum of integers: 20
Sum of doubles: 5.4
```

```java
class car{
  private String make;
  private String model;
  private int year;
  public car(String make, String model, int year){
    this.make=make;
    this.model=model;
    this.year=year;
  }
  public void printcardetails(){
    System.out.println("Make: "+make);
    System.out.println("Model: "+model);
    System.out.println("Year: "+year);
  }
}
public class R192211567{
  public static void main(String[] args){
    car mycar=new car("honda", "amaze", 2017);
    mycar.printcardetails();
  }
}
```

```
Your INPUT go's here! Give only values. do not give like a=10
```

```
Make: honda
Model: amaze
Year: 2017
```

```java
    }

    @Override
    public void eat() {
        System.out.println(name + " is eating seeds or insects.");
    }

    @Override
    public void reproduce() {
        System.out.println(name + " is laying eggs.");
    }
}

public class R192211567 {
    public static void main(String[] args) {
        Mammal lion = new Mammal("Lion", "Savanna");
        Reptile snake = new Reptile("Snake", "Forest");
        Bird eagle = new Bird("Eagle", "Sky");

        lion.eat();
        lion.reproduce();

        snake.eat();
        snake.reproduce();

        eagle.eat();
        eagle.reproduce();
    }
}
```

```
Lion is eating meat or plants.
Lion is giving birth to live young.
Snake is eating insects or other animals.
Snake is laying eggs.
Eagle is eating seeds or insects.
Eagle is laying eggs.
```

```
18      void play() {
19          System.out.println("Playing action game: " + title + " with difficulty " + difficulty);
20      }
21  }
22
23  class PuzzleGame extends Game {
24      int puzzles;
25
26      PuzzleGame(String title, int puzzles) {
27          super(title, "Puzzle");
28          this.puzzles = puzzles;
29      }
30
31      void play() {
32          System.out.println("Playing puzzle game: " + title + " with " + puzzles + " puzzles.");
33      }
34  }
35
36  public class R192211567 {
37      public static void main(String[] args) {
38          ActionGame action = new ActionGame("Warrior's Quest", 5);
39          PuzzleGame puzzle = new PuzzleGame("Mystery of the Ancients", 10);
40
41          action.play();
42          puzzle.play();
43      }
44  }
45
```

```
46          super(title, author);
47          this.duration = duration;
48      }
49
50      public int getDuration() {
51          return duration;
52      }
53
54      public void borrow() {
55          System.out.println("DVD borrowed: " + getTitle());
56      }
57
58      public void returnItem() {
59          System.out.println("DVD returned: " + getTitle());
60      }
61  }
62
63  public class R192211567 {
64      public static void main(String[] args) {
65          Book book = new Book("Harry Potter", "J.K. Rowling", 309);
66          DVD dvd = new DVD("The Lord of the Rings", "Peter Jackson", 241);
67
68          book.borrow();
69          book.returnItem();
70
71          dvd.borrow();
72          dvd.returnItem();
73      }
74  }
```

```
        return a - b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
    public int divide(int a, int b) {
        return a / b;
    }
}
class ScientificCalculator extends Calculator {
    @Override
    public int multiply(int a, int b) {
        return (int) Math.pow(a, b);
    }
}
public class R192211567{
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Basic Calculator Results:");
        System.out.println("Addition: " + calc.add(7,5));
        System.out.println("Subtraction: " + calc.subtract(9,6));
        System.out.println("Multiplication: " + calc.multiply(5,3));
        System.out.println("Division: " + calc.divide(7,5));

        ScientificCalculator sciCalc = new ScientificCalculator();
        System.out.println("\nScientific Calculator Results:");
        System.out.println("Multiplication: " + sciCalc.multiply(8, 3));
    }
}
```