Class Diagram

```
Class Car {
    String brand;
    String model;
    int engineCC;

    startEngine();
    stopEngine();
    accelerate();
    brake();
}
```

<>

Car
+ brand : String
+ model : String
+ engineCC : int
− startEngine() : void
− stopEngine() : void
− accelerate() : void
− brake() : void

} Class Name

} characters/ Variables

} Behaviours/ methods

Generic Class

<>

ClassName
(+) var1 : dataType
(−) var2 : dataType
(+) method1() : dT
(−) method2() : dT

Associations

```
        Associations
        /          \
  Class          Object
Association    Association
    |          /    |     \
Inheritance  Simple  Aggregation  Composition
            Association
                  _____/
                      |
                  Composition
```

Inheritance

→ (is-a) relationship
↳ Denoted by (⟶)

```
Class A {
    method1();
}

Class B : public A {
    method2();
}

main() {
    B* b = new B();
    b→method1();
    b→method2();
}
```

```
     Animal
    /   |   \
  Cow  Tiger  Human
```

Animal
↑ "is-a"
| Cow |

Composition

→ (has-a) relationship
↳ Denoted by (⟶)

Simple Association

| Arjun | → "has a" | House |

Aggregation

```
              Bed
             /
Sofa — Room
             \
              Chair
```

Container

Composition

Wheels — Chair — Arms
Chair — Seat

Chair
Other things
Wheels Seat Arms

Composition in Code

```
Class A {
    method1();
}

Class B {
    A * a;
    B() {
        a = new A();
    }
    method2();
}
```

```
main() {
    B * b = new B();
    b → method2();
    b → a → method1();
}
```

Sequence Diagram

A → B
Communication/
Interaction

① Representing object:

A   B   C   D   - - -

② Lifeline

A   B
C

② Activation Bar

A   B
Active   Active

④ Messages

Async   Sync

A   B
<<message>>
<<response>>
Sync

A   B
<<message>>
<<message>>

⑤ Create & Destroy Message

Create →

A

<<Create>>
B

Destroy →

A   B

<<Destroy>> ✗

⑥ Lost & Found Msg

lost   found

A
<<LOST>> →⊙

⊙ <<found>>
A

ATM Transaction Flow

Use Case

① 👤 ---> [ATM]
   (acc, amount)

② [ATM] ---> [Transaction]

① Verify ATM pin
③ verify Account
④ Tell Cash Dispenser
   for money.

Sequence Diagram

① Use Case
② Objects
        → ATM
        → User
        → Transaction
        → Account
        → Cash Dispenser

③ Draw Diagram

alt :   (If-else)
Option :   (If)
loop :   for/while