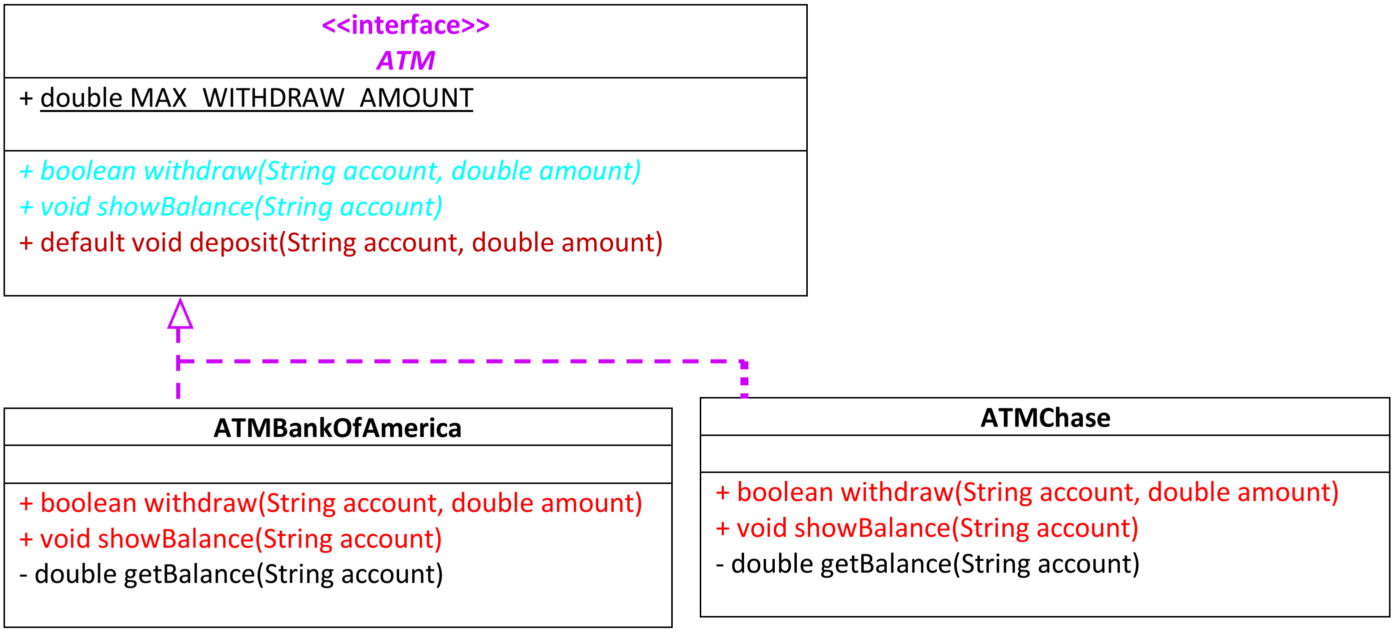
CIS 2168 Fall 2022 Week 1 Inheritance Hierarchies Related to Interface

ATM Inheritance Hierarchy


<<interface>>
ATM
+ double MAX_WITHDRAW_AMOUNT
+ boolean withdraw(String account, double amount)
+ void showBalance(String account)
+ default void deposit(String account, double amount)
ATMBankOfAmerica
+ boolean withdraw(String account, double amount)
+ void showBalance(String account)
- double getBalance(String account)
ATMChase
+ boolean withdraw(String account, double amount)
+ void showBalance(String account)
- double getBalance(String account)

<<interface>>: the construct ATM is an interface. The interface name is marked in *italic.*

----- dashed line: a class implements an interface

The *withdraw(…) method* in **ATM interface** is *abstract* and marked in *italic*.

The deposit(…) method in **ATM interface** is default method. There is no special notation in the UML diagram.

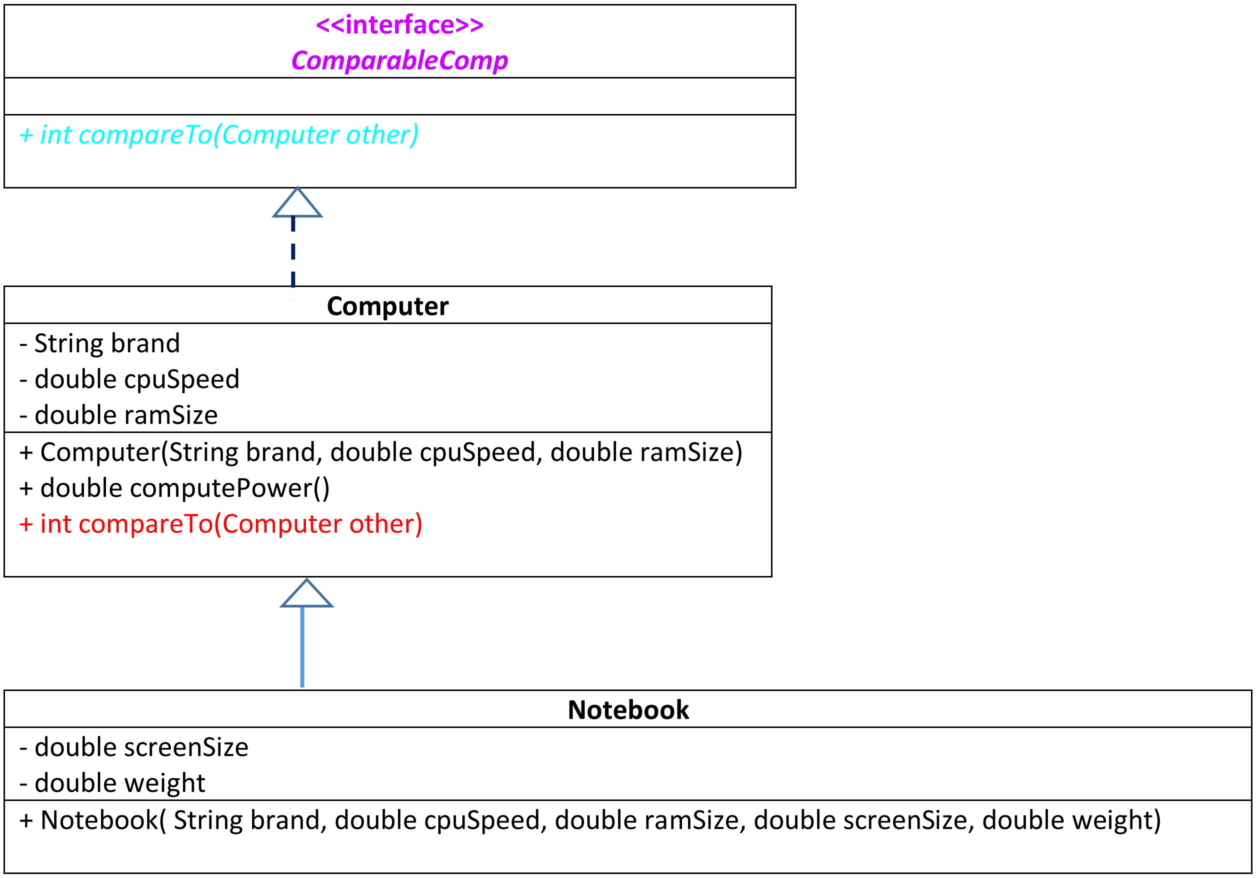
The withdraw(…) method shows method overriding (concrete method overrides abstract method) in Java.

+ means public; **-** means private

Underlined means static method or static data field

Continue to Page 2

Simplified Computer Inheritance Hierarchy with Interface ComparableComp



<<interface>>
ComparableComp

+ int compareTo(Computer other)

Computer

- String brand
- double cpuSpeed
- double ramSize

+ Computer(String brand, double cpuSpeed, double ramSize)
+ double computePower()
+ int compareTo(Computer other)

Notebook

- double screenSize
- double weight

+ Notebook(String brand, double cpuSpeed, double ramSize, double screenSize, double weight)

<<interface>>: the construct ComparableComp is an interface. The interface name is marked in *italic.*

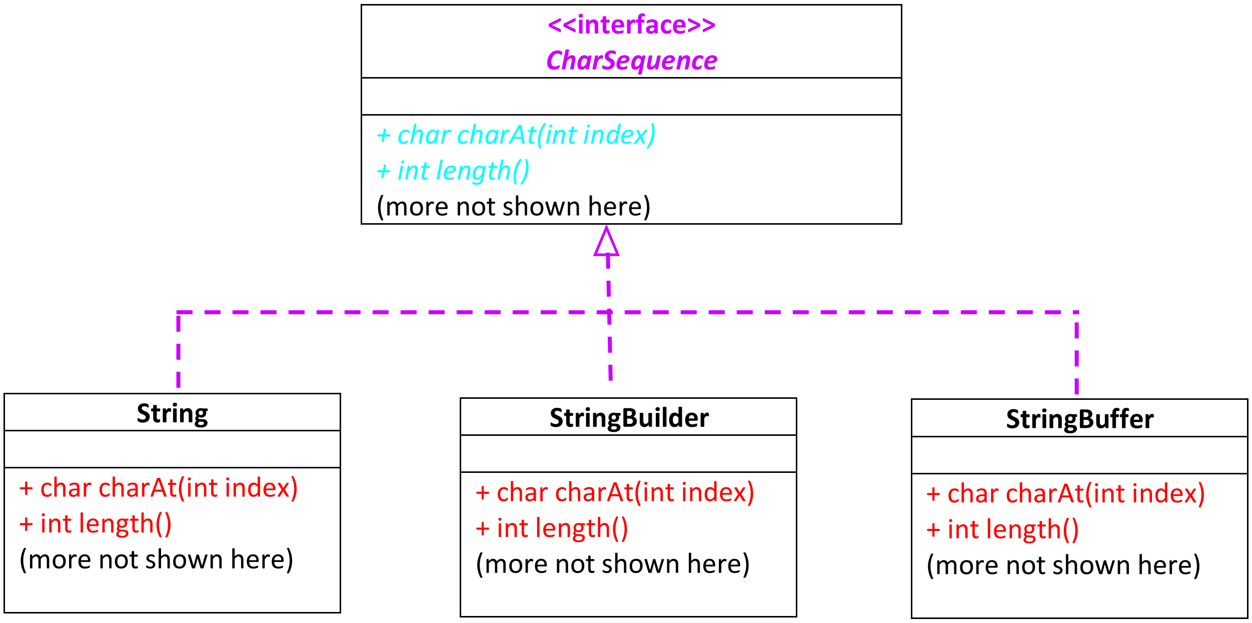
----- dashed line: a class implements an interface

The *compareTo(…) method* in **ComparableComp interface** is *abstract* and marked in *italic*.

The compareTo(…) method shows method overriding (concrete method overrides abstract method) in Java.
+ means public; **-** means private

Continue to Page 3

CharSequence Inheritance Hierarchy in Java API 11



`CharSequence` interface

- A `CharSequence` is a readable sequence of `char` values.
- This interface provides uniform, read-only access to many different kinds of `char` sequences.

Classes that implement CharSequence interface:

- **`String`** – Immutable a sequence of characters
- **`StringBuilder`**
  - A mutable sequence of characters.
  - This class provides an API compatible with `StringBuffer`, but with no guarantee of synchronization.
- **`StringBuffer`**
  - A mutable sequence of characters that supports synchronization for concurrent thread execution

<<interface>>: the construct CharSequence is an interface. The interface name is marked in *italic.*

----- dashed line: a class implements an interface

The *charAt(…), length() methods* in **CharSequence interface** are *abstract* and marked in *italic*.

The charAt(…), length() methods in classes **`String, StringBuilder, StringBuffer`** show method overriding (concrete method overrides abstract method) in Java.

+ means public;

- means private