Programming Contests in November

Everyone is welcome to participate in these contests.

The first programming contest will take place in CSEB 1004 on Friday November 5 from 14:30 until 15:30.

More details about the programming contests can be found at the URL `www.cse.yorku.ca/acm`.

Check06C

Number of students enrolled in the course: 209

Number of students that eChecked Check05C: 17 (8%)

Check06C

Number of students enrolled in the course: 198

Number of students that eChecked Check06C: 18 (9%)

Question

The class `Fraction` has two attributes. What are their names?

Classes and Objects Revisited

Question

The class `Fraction` has two attributes. What are their names?

Answer

`numerator` and `denominator`.

Classes and Objects Revisited

Question

The class `Fraction` has two attributes. What are their names?

Answer

`numerator` and `denominator`.

Question

What are their types?

Classes and Objects Revisited

Question

The class `Fraction` has two attributes. What are their names?

Answer

`numerator` and `denominator`.

Question

What are their types?

Answer

`long` and `long`.

Question

Draw the UML diagram for the `Fraction` class and its attributes.

Question

Draw the UML diagram for the `Fraction` class and its attributes.

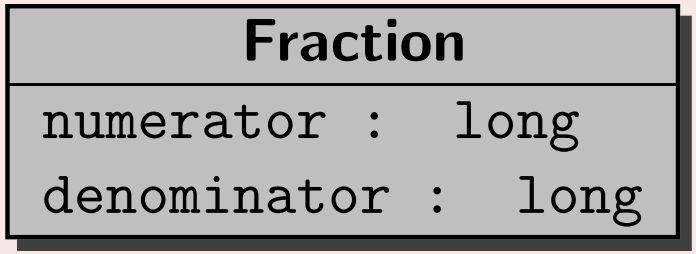
Question

Draw the UML diagram for the `Fraction` class and its attributes.

Question

Draw the UML diagram for the `Fraction` class and its attributes.

Answer



Question

What are the accessors of the class `Fraction`?

Question

What are the accessors of the class `Fraction`?

Answer

`getNumerator` and `getDenominator`.

Classes and Objects Revisited

Question

What are the accessors of the class `Fraction`?

Answer

`getNumerator` and `getDenominator`.

Question

What are the mutators of the class `Fraction`?

Question

What are the accessors of the class `Fraction`?

Answer

`getNumerator` and `getDenominator`.

Question

What are the mutators of the class `Fraction`?

Answer

`setNumerator` and `setDenominator`.

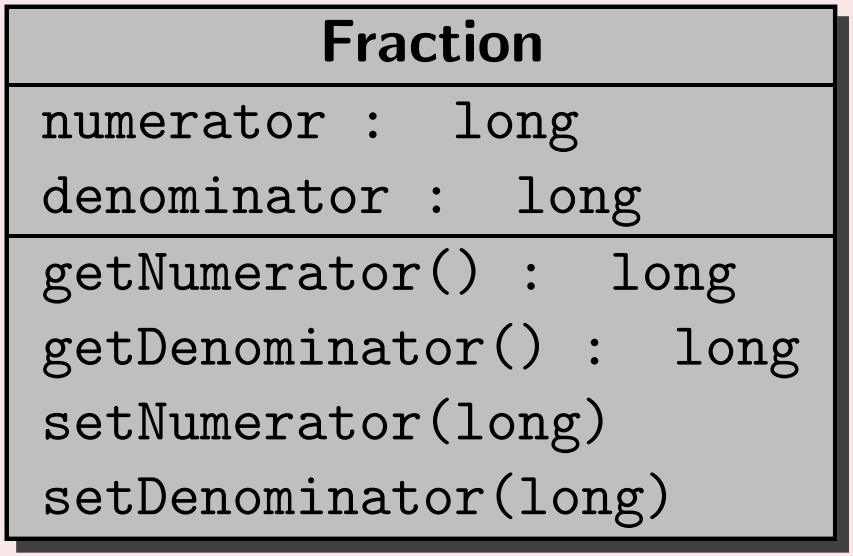
Question

Draw the UML diagram for the `Fraction` class, its attributes, its accessors and its mutators.

Question

Draw the UML diagram for the `Fraction` class, its attributes, its accessors and its mutators.

Answer



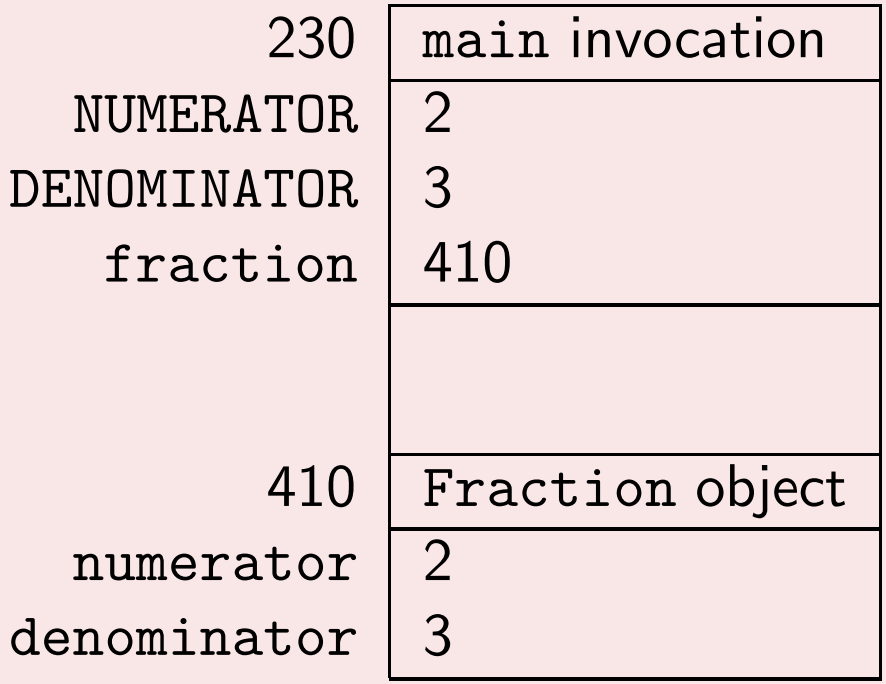
Question

Consider the following body of a `main` method.

```
final int NUMERATOR = 2;
final int DENOMINATOR = 3;
Fraction fraction = new Fraction(NUMERATOR,
   DENOMINATOR);
```

Draw the memory diagram depicting memory at the end of the third line of code.

Answer


230
main invocation
NUMERATOR
2
DENOMINATOR
3
fraction
410
410
Fraction object
numerator
2
denominator
3

Aggregation (Chapter 8)

CSE 1020

November 3, 2010

Combining Data

Combine simple data into more complex data.

1959	COBOL	
1972	C	structures
1979	ML	records
1995	Java	classes

Aggregate

Definition

A class is called an *aggregate* if it has at least one attribute whose type is not primitive.

Example

The class `Stock` of the package `type.lib` is an aggregate because it has an attribute named `symbol` of type `String`.

The class `Investment` of the package `type.lib` is an aggregate because it has an attribute named `stock` of type `Stock`.

The class `Fraction` of the package `type.lib` is not an aggregate because all its attributes are of primitive type.

Aggregation

Definition

Aggregation is a binary relation on classes. The pair (A, P) of classes is in the aggregation relation if class A (aggregate) has an attribute of type P (part).

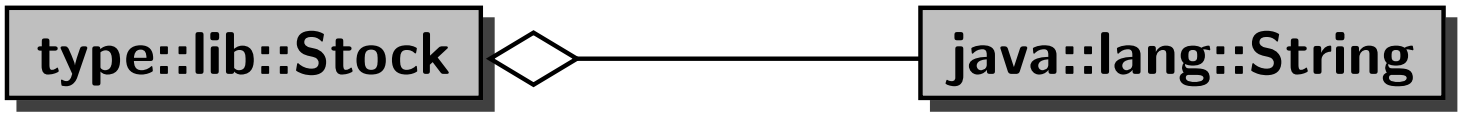
The aggregation relation is also known as the has-a relation. Instead of saying that (A, P) is in the aggregation relation, we often simply say that A has-a P.

Example

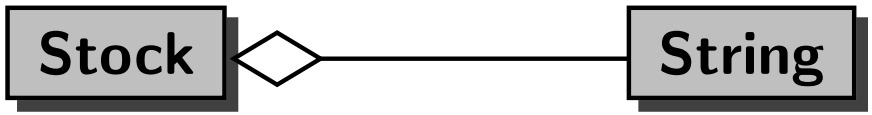
`Stock` has-a `String`.

`Investment` has-a `Stock`.

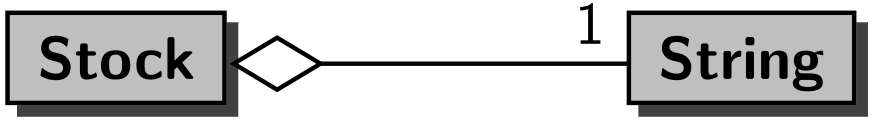
UML Diagrams



UML Diagrams



UML Diagrams



UML Diagrams

`Stock` Object

Question

How do you create a `Stock` object with symbol `"HR.A"`?

Stock Object

Question

How do you create a Stock object with symbol "HR.A"?

Answer

```
String symbol = new String("HR.A"); // "HR.A"
Stock stock = new Stock(symbol);
```

Stock Object

Question

How do you create a Stock object with symbol "HR.A"?

Answer

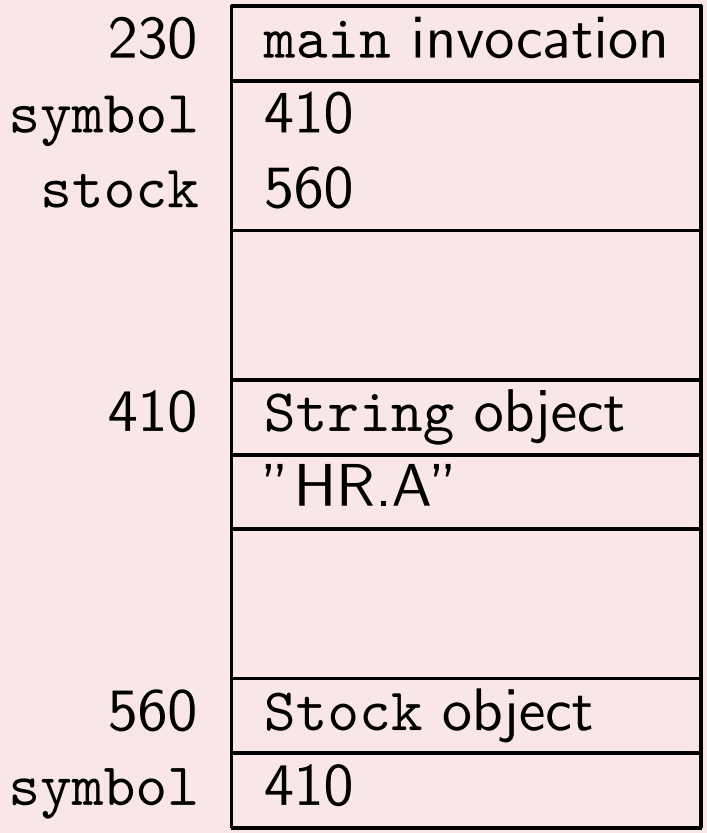
```
String symbol = new String("HR.A"); // "HR.A"
Stock stock = new Stock(symbol);
```

Question

Draw the memory diagram depicting memory at the end of the second line.

Stock Object

Answer



Investment Object

Question

How do you create a `Investment` object with three shares of HR.A stock, each of value 10.00?

Investment Object

Question

How do you create a `Investment` object with three shares of HR.A stock, each of value 10.00?

Answer

```
String symbol = new String("HR.A"); // "HR.A"
Stock stock = new Stock(symbol);
int number = 3;
double value = 10.00;
Investment investment = new Investment(stock, number,
   value);
```

Investment Object

Question

How do you create a `Investment` object with three shares of HR.A stock, each of value 10.00?

Answer

```
String symbol = new String("HR.A"); // "HR.A"
Stock stock = new Stock(symbol);
int number = 3;
double value = 10.00;
Investment investment = new Investment(stock, number,
   value);
```

Question

Draw the memory diagram depicting memory at the end of the fifth line.

Investment Object

230	main invocation
symbol	410
stock	560
number	3
value	10.00
investment	812
410	String object
	"HR.A"
560	Stock object
symbol	410
812	Investment object
stock	560
quantity	3
bookValue	10.00

Accessors

Question

Create a random `Investment` object and print its stock symbol.

Accessors

Question

Create a random `Investment` object and print its stock symbol.

Answer

```
Investment investment = Investment.getRandom();
Stock stock = investment.getStock();
String symbol = stock.getSymbol();
output.println(symbol);
```

Accessors

Question

Create a random `Investment` object and print its stock symbol.

Answer

```
Investment investment = Investment.getRandom();
Stock stock = investment.getStock();
String symbol = stock.getSymbol();
output.println(symbol);
```

Answer (shorter)

```
Investment investment = Investment.getRandom();
output.println(investment.getStock().getSymbol());
```

Accessors

Question

Create a random `Investment` object and print its stock symbol.

Answer

```
Investment investment = Investment.getRandom();
Stock stock = investment.getStock();
String symbol = stock.getSymbol();
output.println(symbol);
```

Answer (shorter)

```
Investment investment = Investment.getRandom();
output.println(investment.getStock().getSymbol());
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Accessors

Answer

230	main invocation
investment	678
stock	
symbol	
234	String object
	"HR.Z"
456	Stock object
symbol	234
678	Investment object
stock	456
quantity	8
bookValue	25.50

Accessors

Question

Draw the memory diagram depicting memory at the end of the third line.

Accessors

Answer

230	main invocation
investment	678
stock	456
symbol	234
234	String object
	"HR.Z"
456	Stock object
symbol	234
678	Investment object
stock	456
quantity	8
bookValue	25.50

Mutators

Question

Create a random `Investment` object and set its stock symbol "HR.B".

Mutators

Question

Create a random `Investment` object and set its stock symbol "HR.B".

Answer

```
Investment investment = Investment.getRandom();
Stock stock = investment.getStock();
stock.setSymbol("HR.B");
```

Mutators

Question

Create a random `Investment` object and set its stock symbol "HR.B".

Answer

```
Investment investment = Investment.getRandom();
Stock stock = investment.getStock();
stock.setSymbol("HR.B");
```

Answer (shorter)

```
Investment investment = Investment.getRandom();
investment.getStock().setSymbol("HR.B"));
```

Mutators

Question

Create a random `Investment` object and set its stock symbol "HR.B".

Answer

```
Investment investment = Investment.getRandom();
Stock stock = investment.getStock();
stock.setSymbol("HR.B");
```

Answer (shorter)

```
Investment investment = Investment.getRandom();
investment.getStock().setSymbol("HR.B"));
```

Question

Draw the memory diagram depicting memory at the end of the second line (of the longer answer).

Mutators

Answer

230	`main` invocation
`investment`	678
`stock`	456
234	`String` object
	"HR.Z"
456	`Stock` object
`symbol`	234
678	`Investment` object
`stock`	456
`quantity`	8
`bookValue`	25.50

Mutators

Question

Draw the memory diagram depicting memory at the end of the third line.

Accessors

230	main invocation
investment	678
stock	456
234	String object
	"HR.Z"
456	Stock object
symbol	890
678	Investment object
stock	456
quantity	8
bookValue	25.50
890	String object
	"HR.B"

How to Copy an Object?

We will show three ways to copy an object:

- create an alias,
- create a shallow copy, and
- create a deep copy.

The created copies are fundamentally different.

How to Create an Alias?

Question

How to create an alias of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

How to Create an Alias?

Question

How to create an alias of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment alias = investment;
```

How to Create an Alias?

Question

How to create an alias of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment alias = investment;
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Alias

230	`main` invocation
`investment`	678
`alias`	
234	`String` object
	"HR.Z"
456	`Stock` object
`symbol`	234
678	`Investment` object
`stock`	456
`quantity`	8
`bookValue`	25.50

Alias

Question

Draw the memory diagram depicting memory at the end of the second line.

Alias

230	main invocation
investment	678
alias	678
234	String object
	"HR.Z"
456	Stock object
symbol	234
678	Investment object
stock	456
quantity	8
bookValue	25.50

How to Create a Shallow Copy?

Question

How to create a shallow copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

How to Create a Shallow Copy?

Question

How to create a shallow copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment shallowCopy = new Investment(
   investment.getStock(),
   investment.getQty(),
   investment.getBookValue());
```

How to Create a Shallow Copy?

Question

How to create a shallow copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

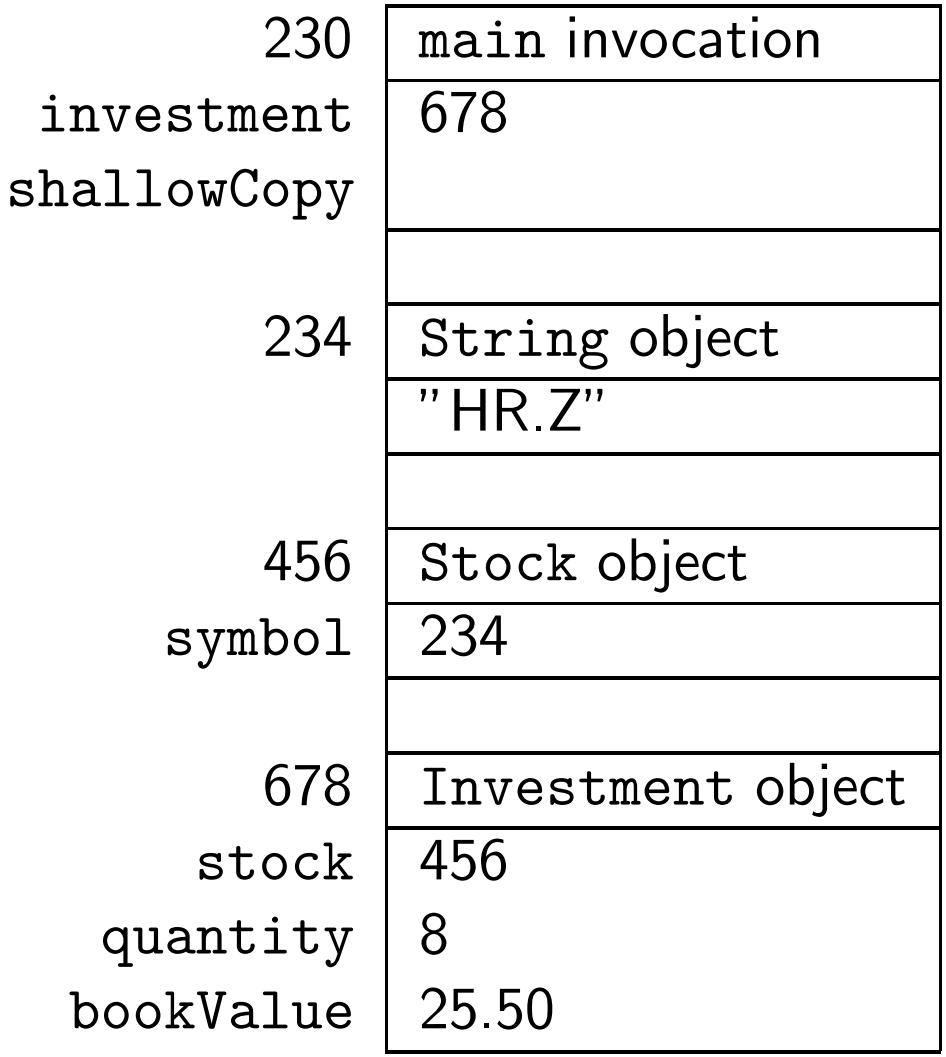
Answer

```
Investment shallowCopy = new Investment(
    investment.getStock(),
    investment.getQty(),
    investment.getBookValue());
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Shallow Copy



Shallow Copy

Question

Draw only those blocks of the memory diagram that change when reaching the end of the fifth line.

Shallow Copy

230	`main` invocation
`investment`	678
`shallowCopy`	1046
1046	`Investment` object
`stock`	456
`quantity`	8
`bookValue`	25.50

How to Create a Deep Copy?

Question

How to create a deep copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

How to Create a Deep Copy?

Question

How to create a deep copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment deepCopy = new Investment(
   new Stock(
     new String(investment.getStock().getSymbol())),
   investment.getQty(),
   investment.getBookValue());
```

How to Create a Deep Copy?

Question

How to create a deep copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Answer

```
Investment deepCopy = new Investment(
   new Stock(
     new String(investment.getStock().getSymbol())),
   investment.getQty(),
   investment.getBookValue());
```

Question

Draw the memory diagram depicting memory at the end of the first line.

Deep Copy

230	main invocation
investment	678
deepCopy	
234	String object
	"HR.Z"
456	Stock object
symbol	234
678	Investment object
stock	456
quantity	8
bookValue	25.50

Shallow Copy

Question

Draw only those blocks of the memory diagram that change when reaching the end of the fifth line.

Deep Copy

230	`main` invocation
`investment`	678
`deepCopy`	1046
1046	`Investment` object
`stock`	1234
`quantity`	8
`bookValue`	25.50
1234	`Stock` object
`symbol`	1288
1288	`String` object
	"HR.Z"

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Answer

No. It only wastes memory.

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Answer

No. It only wastes memory.

Question (revisted)

How to create a deep copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Deep Copy

Question

Recall that `String` objects are immutable. Is there any point of having two identical `String` objects in memory?

Answer

No. It only wastes memory.

Question (revisted)

How to create a deep copy of the following `Investment` object?

```
Investment investment = Investment.getRandom();
```

Answer (improved)

```
Investment deepCopy = new Investment(
   new Stock(investment.getStock().getSymbol()),
   investment.getQty(),
   investment.getBookValue());
```

Composition

Composition is a special type of aggregation. The aggregate A and its part P form a composition if "A owns P", that is, each object of type A has exclusive access to its attribute of type P.

The designer and the implementer of a class determine whether an aggregation is a composition.

Java does not provide any special language constructs for implementing compositions. The constructors, accessors and mutators are implemented in a particular way (the details will be covered in CSE1030).