Unit 3: Class Creation
Abstraction and Program Design

Aligned to:

AP Computer Science A Course and Exam Description

Effective Fall 2025

Topic 3.1

https://longbaonguyen.github.io

What is Abstraction?

- abstraction: reducing complexity by hiding the details you do not need right now.
- You have used it all year without naming it. When you write:

```
double r = Math.sqrt(2);
```

- you do not know or care how Java computes a square root. You only need to know what it gives back.
- Abstraction lets you build programs larger than you could keep in your head all at once.

Two Kinds to Name

- procedural abstraction: using a method for what it does, without knowing how it does it.
- You call it by name. Someone else wrote the body.
- data abstraction: separating what a type is from how it is actually stored.
- A Deck of cards is a deck whether it is stored in an array or an ArrayList. The client does not need to know which.
- In both cases you give something a name and keep the details out of the way.

Procedural Abstraction

- Here is the same computation, written out three times:

```
int sum1 = 0;
for (int s : ada)  sum1 += s;
double avgAda = (double) sum1 / ada.length;
```

- ...and again for ben, and again for chen. Now change the rule and you must find and fix all three.
- Pull it into one method and the duplication is gone:

```
public static double average(int[] a)
```

- Parameters are what make a method general. Without them you would need a separate method per array.

Data Abstraction

- Suppose you are given a Deck class and told only this:

```
void shuffle()
String deal()
int cardsRemaining()
```

- You can write an entire card game against that, without ever opening Deck.java.
- Later the author swaps the internal array for an ArrayList. Your game still compiles and still runs.
- The internals could change because the client only ever depended on the method names and what they do.

Attributes

- The CED gives these terms precise meanings, and they are testable:
- An ATTRIBUTE is a type of data abstraction defined in a class, outside any method or constructor.
- An INSTANCE VARIABLE is an attribute whose value is unique to each object of the class.
- A CLASS VARIABLE is an attribute shared by all instances of the class.
- So attribute is the design word. Instance variable and class variable are the two kinds you actually declare.

Design Before You Type

- Given a problem in English, the design work happens before any Java.
- Read the description and look for two things:
- NOUNS become classes and attributes: the things the program knows.
- VERBS become behaviors: the things the program does.
- This is not a rule that always works, but it is a reliable place to start.

A Vending Machine

- "The machine holds a number of items, each with a price. It accepts coins, tells the customer how much they have inserted, dispenses an item if enough money was paid, and returns change."
- Attributes (what it knows): the items it holds, each item price, money inserted so far
- Behaviors (what it does): insertCoin, amountInserted, dispense, giveChange, restock
- Notice that no Java has been written yet, and the class is already designed.

Representing a Design

- The CED asks you to represent a design in natural language or in a diagram. Both are fine. A simple box works:

```
VendingMachine
------------------------
items, prices, inserted
------------------------
insertCoin(int cents)
dispense(int slot)
giveChange()
```

- Name on top, attributes in the middle, behaviors at the bottom.

This Is FRQ 2

- Question 2 of the free response is Class Design. It hands you a specification in English and asks for a class.
- The translation is always the same:
- attributes → private instance variables
- behaviors → methods
- the values needed to start → the constructor
- Students who lose points here usually started typing before they decided what the class knows.

Your Turn

- "A library book has a title, an author, and a status showing whether it is checked out. A book can be checked out, returned, and asked whether it is available. The library can also be asked to print the title and author together."
- Before writing any code:
- 1) List the attributes.
- 2) List the behaviors, with the parameters each one needs.
- 3) Decide which values the constructor must receive.
- Compare your design with a partner. More than one good answer exists here.

References

1) Building Java Programs: A Back to Basics Approach by Stuart Reges and Marty Stepp

2) Runestone CSAwesome Curriculum: https://runestone.academy/runestone/books/published/csawesome/index.html

For more tutorials/lecture notes in Java, Python, game programming, artificial intelligence with neural networks:

https://longbaonguyen.github.io