Unit 3: Class Creation
Impact of Program Design

Aligned to:

AP Computer Science A Course and Exam Description

Effective Fall 2025

Topic 3.2

https://longbaonguyen.github.io

Programs Change Things

- Every program you write does something in the world. Some of what it does, you designed. Some of it you did not.
- A program can have effects on society, on the economy, and on culture. Some are beneficial, some are harmful, and many were not predicted when the code was written.
- Unintended consequences are common rather than rare, so part of designing a program is asking what it might do that you did not intend.

Reliability

- A program is reliable when it does what is expected, under the conditions it was built for, without failing.
- Reliability comes from testing with a variety of inputs and conditions, especially the inputs you did not have in mind while writing the code.
- Code that works on the example given in the assignment has not yet been shown to work in general.

Testing for Edge Cases

- The interesting inputs are almost always at the edges. For a method that takes an array, try:
- an empty array (length 0)
- a single element (length 1)
- all values the same
- negative values, and zero
- the largest value the type can hold
- Most bugs that survive into a finished program are hiding in one of these cases.

The Stakes Differ

- Consider one bug: an average computed with integer division, so it silently truncates.
- In a class project: a wrong number on the screen.
- In a bank’s interest calculation: money quietly created or destroyed, across every account, every day.
- In a medical infusion pump: a dose.
- It is the same bug in each case. How much care a program deserves depends on what it affects.

Using Others' Code

- You will often find code that solves your problem. Whether you may use it is a separate question from whether it works.
- One rule to remember:
- Being able to SEE code does not mean you are allowed to USE it.
- Code published as open source and free to use tells you what you may do with it. Code that is not open source requires you to obtain permission, and often to purchase it, before integrating it into your program.

Three Sources

- You want to sell an app. May you paste in code from:
- 1) a repository with an MIT or Apache license?
- Yes. Read the license and follow its attribution terms.
- 2) a repository with no license file at all?
- No. Obtain permission first, and expect to have to purchase it.
- 3) an answer posted on a Q&A site?
- It depends on that site’s terms, which usually require attribution.
- Most people guess all three, which can be an expensive mistake.

Writing Code With AI

- NOT ASSESSED: artificial intelligence does not appear in the Course Framework. The next few slides are included because the skill is useful, not because the exam asks for it.
- AI tools produce working Java quickly. That changes how fast you write code. It does not change who is responsible for it. They are good at boilerplate, half-remembered syntax, and explaining an error message.
- What it is unreliable at: knowing your requirements, handling the edge cases you did not mention, and knowing when it is wrong.
- It will produce confident, well-formatted code that is wrong, and it will not warn you when that happens.

What Can Go Wrong

- Ask an AI for "a Java method that returns the average of an int array" and you will very often get something like:

```
public static double average(int[] a) {
    int sum = 0;
    for (int x : a) sum += x;
    return sum / a.length;
}
```

- Three defects: integer division truncates the answer, an empty array divides by zero, and a long array can overflow sum. The code compiles and looks correct.

You Ship It, You Own It

- If you did not write a line of code, you still have to be able to explain what it does before you use it.
- College Board states this directly for AP Computer Science Principles: it is the student’s responsibility to review and understand any code co-written with AI tools, and to make sure it works.
- The same principle applies here. Saying that an AI tool wrote the code does not explain why it failed.

Your Turn

- In pairs:
- 1) Ask an AI tool for a Java method that returns the MEDIAN of an int array.
- 2) Do not run it. Read it, and write down every input you think would break it.
- 3) Now run it against those inputs.
- 4) Write two sentences: what did it get wrong, and who would have been responsible if this had shipped?
- Bring the code and your notes to the next class.

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