# Lecture Summary: Introduction to Hackathons and Object-Oriented Programming

Generated by LectureMate

February 22, 2025

## 1 Introduction

The lecture began with an introduction to the Hackler Boro event, which is the 11th iteration of an annual 24-hour hackathon scheduled for March 1st and 2nd, 2025. This event is organized by Comstock in collaboration with other societies and is considered a significant opportunity for students to enhance their CVs and gain experience in hackathons.

### 1.1 Event Details

- **Date:** March 1-2, 2025
- **Format:** 24-hour hackathon
- **Participation:** Open to all students, no prior experience required
- **Prizes:** Sponsored by major companies (names to be announced)

## 2 Course Updates

The instructor provided updates on the course, including the release of Assignment 1 and feedback from previous sessions. Students were encouraged to engage with lab sessions and Piazza for support and resources.

### 2.1 Assignment 1

- Released today; students are advised to start early.
- Feedback indicated a mix of preferences for live coding versus slides.
- Most students are enjoying the course, with a few finding it challenging.

## 3 Object-Oriented Programming Concepts

The latter part of the lecture focused on object-oriented programming (OOP) principles, including classes, objects, and methods.

### 3.1 Classes and Objects

- **Class:** A blueprint for creating objects (e.g., a Car class).
- **Object:** An instance of a class that contains state (attributes) and behavior (methods).
- **Constructor:** A special method used to initialize objects.

### 3.2 State and Behavior

- Objects have **state** (instance variables) and **behavior** (methods).
- Example: A Car object may have attributes like speed and fuel level, and methods like startEngine() and accelerate().

### 3.3 Memory Management

The lecture discussed how objects are stored in memory and how references work in Java. Key points include:

- Objects are created in memory using the `new` keyword.
- References point to the memory location of an object.
- Copying references does not create a new object; it simply points to the same memory location.

### 3.4 Equality in Objects

The difference between `==` and `.equals()` was explained:

- `==` checks if two references point to the same object in memory.
- `.equals()` checks if two objects have the same value/state.

## 4 Conclusion

The lecture concluded with a reminder about the importance of participating in hackathons and engaging with course materials. Students were encouraged to ask questions and utilize office hours for additional support.