Object-Oriented Programming Educational Materials

> **CS 165 Course Materials Repository**
> A comprehensive educational content repository with professional markdown styling and PDF export capabilities.

🎯 Repository Overview

This repository contains **Object-Oriented Programming in C++** course materials with a sophisticated **VS Code-based authoring system**. It's designed as part of a series of educational repositories, featuring consistent styling, professional PDF export, and comprehensive educational content organization.

📁 Repository Structure

```
object-oriented-programming/
├── .vscode/                    # 🎯 MAIN DRIVER - Complete styling system
│   ├── settings.json           # VS Code configuration & Git automation
│   ├── markdown-styles.css     # 1,200+ lines of Bootstrap-inspired styles
│   ├── markdown.code-snippets # 100+ educational content snippets
│   ├── markdown-styles-demo.md # Complete style showcase
│   └── assets/                 # Educational character images (Sue & Sam)
├── cpp/                        # Core course content (CS 165)
│   ├── Unit-0/                # Course Overview & Review
│   ├── Unit-1/                # Using Objects (6 chapters)
│   ├── Unit-2/                # Encapsulation & Classes (9 chapters)
│   ├── Unit-3/                # Inheritance (6 chapters)
│   ├── Unit-4/                # Advanced Topics (7 chapters)
│   └── Appendix/              # Reference materials
├── pdf-ocr-obsidian/          # Processing directory (empty)
└── Support files              # README, quick reference, etc.
```

🎨 Main Driver: The `.vscode` Styling System

The **`.vscode` directory** is the heart of this repository - a complete educational content authoring system that transforms basic markdown into professional, styled educational materials.

✨ Key Features

Professional Styling Engine

- **1,200+ lines of CSS** with Bootstrap-inspired callouts
- **18 callout types**: Warning, Tip, Info, Error, Success, Bug, etc.
- **28 code block style variants** matching callout themes
- **Professional table styling** with blue headers and alternating rows
- **Educational character sections** (Sue's Tips, Sam's Corner)

Complete Authoring Workflow

- **100+ VS Code snippets** for rapid content creation
- **WYSIWYG PDF export** - preview matches PDF exactly
- **Dynamic footer system** with course/unit/chapter info
- **No external dependencies** - works with VS Code alone

Educational Enhancements

- **Obsidian-style callouts** for enhanced learning
- **Mathematical expression support** (LaTeX)
- **Interactive collapsible sections**
- **Task lists and definition lists**
- **Character-based learning aids**

🔧 Author Preferences (Mr. Eli's Standards)

- **Line breaks**: Uses \ instead of trailing spaces for clarity
- **Professional presentation**: Bootstrap-inspired without Bootstrap dependency
- **Educational focus**: Callouts, characters, and interactive elements
- **Consistency**: Unified styling across all course materials
- **Portability**: Styling system can be copied to other repositories

📚 Course Content Scope

CS 165: Object-Oriented Programming in C++

Unit 0: Course Foundation

- Course Overview & Objectives
- Review of Programming Fundamentals

Unit 1: Using Objects (6 Chapters)

- 1.0 Design Documents
- 1.1 Defensive Programming
- 1.2 Exception Handling
- 1.3 Structures
- 1.4 Separate Compilation
- 1.5 Function: Advanced Topics

Unit 2: Encapsulation & Classes (9 Chapters)

- 2.0 Encapsulation Design
- 2.1 Building a Class
- 2.2 Class Syntax
- 2.3 Accessors & Mutators
- 2.4 Constructors & Destructors
- 2.5 Static Members
- 2.6 Non-Member Operator Overloading
- 2.7 Friends

- 2.8 Member Operator Overloading

Unit 3: Inheritance (6 Chapters)

- Advanced object-oriented concepts
- Inheritance hierarchies
- Polymorphism implementation

Unit 4: Advanced Topics (7 Chapters)

- Template programming
- Design patterns
- Performance optimization

Appendices

- **Appendix A**: Elements of Style
- **Appendix B**: C++ Reference Guide
- **Appendix C**: Glossary
- **Appendix D**: Index

🚀 Getting Started

For Content Creation:

1. **Open the repository** in VS Code
2. **Install recommended extensions**:
    - Markdown Obsidian Callout
    - Markdown Extended (for PDF export)
3. **Use code snippets** - Type shortcuts like `warning`, `tip`, `codeblock`
4. **Preview in real-time** - What you see is what you export to PDF

For PDF Export:

1. **Ensure `.vscode` folder** is in workspace
2. **Open any markdown file**
3. **Use Command Palette**: `Ctrl+Shift+P` → "Markdown Extended: Export (pdf)"
4. **Result**: Professional PDF with custom footers and styling

For Reusing the Styling System:

1. **Copy the entire `.vscode` folder** to your new repository
2. **Modify CSS variables** in `markdown-styles.css` for your course
3. **Update snippets** in `markdown.code-snippets` as needed
4. **Customize character images** in `.vscode/assets/`

📖 Content Standards

Frontmatter Template

```yaml
---
title: "Chapter Title"
description: "Brief description"
course: "CS 165"
unit: "Unit X"
chapter: "Chapter-Name"
tags:
  - cpp
  - object-oriented-programming
source:
  type: "AI-Generated Draft"
  method: "Mistral OCR"
  original: "source.pdf"
  generated: "2025-08-03 16:15:23"
author: "CS 165 Course Materials"
date: "2025-08-03"
---
```

Callout Examples

```markdown
> [!WARNING]
> Critical information that students must know

> [!TIP]
> Helpful advice for better understanding

> [!EXAMPLE]
> Sample code or usage demonstration
```

Educational Characters

```markdown
> [!DANGER] 💡 **Sue's Tip:**
> ![Sue](.vscode/assets/sue.png){width=50 align=right}
> Practical advice for getting things done efficiently

> [!TLDR] 🤓 **Sam's Corner:**
> ![Sam](.vscode/assets/sam.png){width=50 align=right}
> Technical details and interesting tidbits
```

🔄 Workflow Integration

VS Code Configuration

```
{
    "markdown.styles": ["./.vscode/markdown-styles.css"],
    "markdown.preview.breaks": false,
    "markdown.preview.linkify": true,
    "markdown.preview.typographer": true,
    "gitdoc.enabled": true,
    "gitdoc.autoCommitDelay": 10000,
    "gitdoc.commitMessageFormat": "🤖 AI Commit: ${aiMessage} | ${date}"
}
```

Git Automation

- **Auto-commits** every 10 seconds during editing
- **AI-generated commit messages** with emojis
- **Auto-push** every 30 seconds for backup

🎯 Strategic Value

This repository serves as:

1. **Educational Content Hub** - Complete CS 165 course materials
2. **Styling System Template** - Reusable across multiple repositories
3. **Authoring Workflow** - Professional academic content creation
4. **PDF Generation Engine** - Consistent, branded educational materials
5. **Repository Series Foundation** - Base for other temporary educational repos

📋 Next Steps

- [ ] **Populate README files** in other units/chapters
- [ ] **Complete missing content** in Units 3-4
- [ ] **Add more character interactions** (Sue & Sam)
- [ ] **Create batch PDF export** for entire course
- [ ] **Template system** for new courses

📝 Quick Access

- Style Demo - Complete showcase of all available styles
- Course Overview - CS 165 introduction and structure
- Unit 1 Start - Begin with Using Objects
- Quick Reference - Essential markdown syntax

> ✓ **Repository Status**
>
> ✅ Complete styling system implemented
> ✅ Professional PDF export workflow

✅ Educational content structure established
✅ Consistent authoring standards defined

Last updated: August 6, 2025