SE: Portfolio - Case Studies

Session 8

Functions: Vehicle Dashboard

The automotive industry has seen significant growth in South Africa over the last few years and contributes significantly to the GDP of the country. There are only 22 companies in South Africa involved in the production of cars and commercial vehicles and you've been hired by one of them to be part of their software engineering team.

You've been selected to be in charge of one of the most important aspects of the car: the onboard entertainment system. Given how vital and powerful technology is in our lives, the vehicle dashboard is a major contributing factor when deciding which car to buy. The features offered, the attractiveness of the user interface, how "smart" the system all play a role in the driver experience.

The original software used, called DashMaster, has become outdated and is facing very stiff competition in the industry. You have to design the newest vehicle dashboard software, DashMaster Pro, which will be used for the onboard entertainment system in the next generation of vehicles produced by the company. This case study will involve creating the backend of the system only.

Students are tasked with designing the backend of a vehicle dashboard and infotainment system which leverages the terminal as opposed to the entertainment system and screen found in cars. The purpose of this task is to design all the functions that will be used by drivers. There are several categories each of which has several functions available to the driver. Students must ensure that the system is well documented, easily modifiable and modular.

Session 9

Functions: Fitness Program

Tomasz is a fitness coach committed to assisting his clients in reaching their wellness objectives. His goal is to develop a fitness program that lets users design custom diets and exercise regimens using user-defined features. Tomasz has tried coding the program himself but the code does not allow for users to create personalised programs and he is finding it difficult to extend the system and introduce more features. Exercise instructions should be generated with the help of string handling, and lists should be utilised to monitor progress and meals. Tomasz hopes that with your help, he can create a program that encourages people to live healthier lifestyles. Could you assist Tomasz in creating a workout aid that changes lives and bodies?

- Create user-defined functions for generating personalised workout routines and nutrition plans.
- Utilise lists to track client progress, meals, and fitness goals.
- Employ string handling for workout instructions and client communication.

Topics to be consolidated

- Programming with User-defined Functions
- Capstone Project -Lists, Functions, and String Handling

Session 10

OOP - ZooWonders

Construct a virtual zoo administration system with classes and objects. In the program, define each animal's traits and behaviours by representing it as an object. Provide a lively and instructive environment where guests can engage with virtual animals and discover more about their habitats and habits.

- Create classes for various animal types with attributes and behaviours.
- Instantiate objects for different animals within the virtual zoo.
- Develop a user interface for visitors to interact with the virtual animals ***.

Topics to be consolidated

- Classes
- Objects
- Define the fundamental concepts of OOP, including classes, objects, attributes, and methods.
- Describe the role of an object in Python.
- Create a Python class with its attributes and methods.
- Explain and use the '__init__' method to create a class constructor to initialise object attributes.
- Explain the role of magic methods (e.g., __str__, __eq__) in customising class behaviour.
- Define an instance of your class, manipulate the attributes of the instance, and call its methods.

Session 11

OOP - Quest for the Lost Relics

Your skills are required to create another game! Creating a thrilling treasure hunt game that makes use of inheritance and abstraction is your task. Make a game where various treasure kinds share characteristics and actions that come from an abstract treasure class. This strategy will enable you to add new kinds of treasures in the future, making your game fun and extensible.

- Create an abstract treasure class with common attributes and behaviours.

- Implement inheritance to create specific types of treasures (e.g., gold, gems, artefacts).
- Develop a treasure hunt game where players collect various treasures with distinct properties.

Topics to be consolidated

- Inheritance
- Abstraction
- Explain the concept of encapsulation and use private attributes and methods within your class to implement encapsulation.
- Describe inheritance and its role in OOP.
- Create subclasses that inherit attributes and methods from superclasses.
- Override superclass methods within your subclasses.

Session 12

OOP - The Interactive Odyssey

Put on your author hat and start telling stories. Your goal is to produce an interactive digital storybook with various characters and plots that will captivate readers. Build a system where characters in the story can interact and have different outcomes by using polymorphism and encapsulation. Character objects should contain the interactions between the characters, and polymorphism will enable characters to react differently to the reader's choices, providing a personalised reading experience for each user.

- Create character classes with encapsulated interactions and outcomes.
- Implement polymorphism to allow characters to respond differently to user choices.
- Develop an interactive storybook that offers readers a personalised and engaging narrative.

Topics to be consolidated

- OOP - Polymorphism and Encapsulation
- Define polymorphism and demonstrate its behaviour through method overriding and inheritance.
- Explain the concept of abstraction and how it fits into OOP.
- Apply SOLID principles (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) to class design.

- Analyse how OOP can be used to write DRY code

Session 13 (Shared with DS)

GIT: CodeHaven's Collaborative Challenge

The open-source community is a testament to the collaborative spirit of software development, where ideas flourish and knowledge is freely shared. "CodeHaven" stands as a beacon for this ethos, inviting developers from all walks of life to contribute and learn.

Alex, a developer with a keen interest in mentoring, presents the students with a challenge that serves as a rite of passage into the open-source world. The task is to contribute to "First Contributions," (firstcontributions/first-contributions: ✨Help beginners to contribute to open source projects (github.com)) a welcoming project for novices in the open-source community.

The students' task is to master the tools of the trade: Git and GitHub. They will learn to navigate code repositories, manage updates, and understand the social coding environment that GitHub fosters. Through this hands-on experience, they will not just contribute code, but also craft a narrative of their development journey.

Their contributions will be their legacy, a digital portfolio that speaks to their ability to engage with complex workflows and collaborative projects. It's a chance to turn "CodeHaven" into a showcase of their dedication, skill, and readiness to join the global developer community.

Topics to be consolidated

- Understanding the basics of version control with Git commands
- Navigating and using GitHub for code and profile management
- Employing branching strategies for safe and effective updates
- Engaging in collaborative code reviews with pull requests
- Showcasing projects effectively on GitHub
- Grasping the distributed nature of Git for teamwork
- Exploring alternatives to Git for version control