



**SCHOOL OF
SCIENCE AND
TECHNOLOGY**

PAN-ATLANTIC UNIVERSITY



PAN-ATLANTIC UNIVERSITY

TECHNOLOGY

B.Sc. Computer Science (CSC)

COS 202 – Computer Programming II

Dr. Pius Onobhayedo

ponobhayedo@pau.edu.ng

Sochukwuma Nwokoye

snwokoye@pau.edu.ng



Learning Outcomes

- At the end of this course, students should be able to:
- demonstrate the principles of good programming and structured programming concepts;
- demonstrate string processing, internal searching, sorting, and recursion;
- demonstrate the basic use of OOP concepts: classes, objects, inheritance, polymorphism, data abstraction;
- apply the tools for developing, compiling, interpreting and debugging programs; and
- demonstrate the use of syntax and data objects, operators. Central flow constructs, objects and classes programming, Arrays, methods, Exceptions, Applets and the Abstract, OLE, Persistence, Window Toolkit.



Course Contents

- Review and coverage of advanced object-oriented programming - polymorphism, abstract classes and interfaces; Class hierarchies and program organisation using packages/namespaces; Use of API – use of iterators/enumerators, List, Stack, Queue from API; Searching; sorting; Recursive algorithms; Event-driven programming: event-handling methods; event propagation; exception handling. Applications in Graphical User Interface (GUI) programming.

Course Objectives (Adjusted)

- ❖ Avoid duplication with future CSC 301 (Data Structures)
 - Data structures, OOP, string processing, and the likes will be treated more specifically in 301.
- ❖ This course will lay emphasis on the demonstration of the principles of good programming and structured programming concepts
- ❖ This course is important for your first official internship preparation

...Course Objectives (Adjusted)

- 🏛️ A goal of this course will be to strengthen students' ability to write mature programs for different platforms (Desktop, Web, Mobile, Blockchain), leveraging on industry standard DevOps (development operations).
- 🏛️ In addition, the course is designed to help students master JavaScript, TypeScript and get introduced to Web3 languages like Solidity
- 🏛️ Structured programming will be illustrated in the above context.

Learning Outcomes

- ♥ At the end of this course, the students should
 - be grounded in
 - JavaScript (Classical, ECMAScript 2015 level and beyond)
 - TypeScript as a superset of JavaScript
 - Implement some fundamental concepts in structured and modular programming, using JavaScript
 - get introduced to
 - Solidity as a language for Smart Contract development for Blockchain
 - be grounded in React as a modern-day JavaScript library for Frontend Engineering

...Learning Outcomes

- 🛡️ ...At the end of this course, the students should
- know what it means to write structured software and build for various environments e.g.:
 - Desktop
 - Web
 - Mobile
 - Blockchain
 - Metaverse

...Learning Outcomes

- 🛡️ ...At the end of this course, the students should
- get introduced to know modern DevOps e.g.
 - Version Control
 - Test-driven Development
 - Continuous Integration / Continuous Delivery (CI/CD)
 - Containerized Deployment

Outline

- 🏰 Topic 1: Introduction
 - An overview of where this course fits in the life of a Computer Science major.
 - An overview of modular programming concept
 - An overview of JavaScript: evolution strengths, weaknesses and why the use in this course.
 - Get started with basic development tools for the course
 - VS Code IDE
 - Docker
 - Git
 - Remix and/or Hardhat

Outline

- 🏰 Topic 2: Overview of Platforms
 - Desktop
 - Web
 - Web 1/2 Technology Stack
 - Mobile
 - Native
 - Hybrid
 - DIY

...Outline

- 🏰 Topic 2: Overview of Platforms (cont'd)
 - Blockchain
 - Web3 Technology Stack
 - Emerging Concepts. E.g.
 - Cryptocurrencies
 - Tokens
 - » Non-fungible Tokens (NFT)
 - » Etc.
 - Decentralised finance (DeFi)
 - Decentralised Autonomous Organisations (DAO)
 - Crypto Market making / Automated Market Making (AMM)

...Outline

🏰 Topic 2: Overview of Platforms (cont'd)

– Metaverse

- XR (AR/VR/MR) Concept
- Technology Stacks
 - Oculus Quest
 - HoloLens
 - Vision OS
 - WebXR
 - OpenXR
- NFTs and the Metaverse

...Outline

- 🏰 Topic 3: Classical Programming Concepts in JavaScript
- 🏰 Topic 4: Using ECMAScript 6 standard and beyond
- 🏰 Topic 5: Coding with TypeScript

...Outline

- 🏰 Topic 6: Solutions for the Web
 - Solution Architectures
 - Backend with NodeJS
 - Frontend with ReactJS
- 🏰 Topic 7: Mobile Solutions (Android & iOS)
 - Native codes with React-Native
- 🏰 Topic 8: Desktop Solutions
 - With React-Native-Windows

...Outline



Topic 9: Solutions for the Blockchain

- Backend (Smart Contracts) with Solidity/Python/Rust/Move
- Frontend (Web3) with Web3.js and React/React-Native.

...Outline

- 🛡️ Topic 10: More on DevOps
 - Test-driven Development
 - Jest
 - Continuous Integration / Continuous Delivery
 - Jenkins
 - GitLab or GitHub Actions
 - Distributed Container Deployment
 - Kubernetes