NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Platform Based Development
1.2 Code	:	CS2B1
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS112. Objects-oriented programming I. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Platform-based development focuses on the design and implementation of applications within specific computing environments. This course utilizes the Software Process and Design (SPD) framework to explore the unique constraints and capabilities of Web, Mobile, and Interactive platforms. Students will learn to apply shared architectural concerns and platform-specific APIs to build robust, scalable, and user-centric software solutions.

4. GOALS

- Understand the shared aspects and unique constraints of modern platforms.
- Develop professional applications for Web and Mobile environments.
- Apply interactive computing principles to enhance user experience.
- Design software architectures that leverage platform-specific services.

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

6. TOPICS

Unit 1: Common Aspects: Platforms, APIs and Constraints (5 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Overview of development platforms (i.e., web, mobile, game, robotics, embedded, and interactive). – Input/sensors/control devices/haptic devices – Resource constraints ∗ Computational ∗ Data storage ∗ Memory ∗ Communication – Requirements - security, uptime availability, fault tolerance Reliability Concepts, Reliability Engineering, Security Design and Controls Engineering, Threat Analysis and Security Engineering, Trusted Computing and Privacy Engineering – Output/actuators/haptic devices • Programming via platform-specific Application Programming Interface (API) vs traditional application construction • Programming under platform constraints and requirements (e.g., available development tools, development, security considerations) AI and Hardware Security, Access Control and Applications, Security Mindset and Principles	• List the constraints of mobile programming [List]
Readings : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unit 2: Common Aspects: Platform Languages and Web Programming Patterns (3 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Overview of platform Languages (e.g., Python, Swift, Lua, Kotlin) • Techniques for learning and mastering a platform-specific programming language	• List the characteristics of scripting languages [List] • Describe the three-tier model of web programming [Describe] • Describe how the state is maintained in web programming [Describe]
Readings : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unit 3: Frontend, Frameworks and Web Standards (12 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Web programming languages (e.g., HTML5, JavaScript, Typescript, CSS) • Web platforms, frameworks, or meta-frameworks – Cloud services – API, Web Components • Software as a Service (SaaS) • Web standards such as document object model, accessibility Accessibility and Inclusive Design • Security and Privacy Considerations Computer Crimes and Attacks, Security Laws, Policies and Responsibility	• Compare and contrast web programming with general-purpose programming [Compare] • Describe the differences between Software-as-a-Service (SaaS) and traditional software products [Describe] • Discuss how web standards impact software development [Debate] • Evaluate an existing web application against current web standards [Evaluate]
Readings : [**Flanagan2020**], [**Duckett2014**], [**Fielding:2000:Rest**]	

Unit 4: Mobile Platforms: Development Fundamentals (6 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Development with – Mobile programming languages – Mobile programming environments • Mobile platform constraints – User interface design Understanding the User: Individual goals and interactions with others – Security • Access – Accessing data through APIs Query Construction – Designing API endpoints for mobile apps: pitfalls and design considerations – Network and web interfaces Data Modeling, Fundamentals of Networks and Communications	• Compare mobile programming with general-purpose programming [Compare] • Evaluate the pros and cons of native and cross-platform mobile application development [Evaluate]
Readings : [**Deitel2017**], [**Hellman2013**]	

Unit 5: Mobile Platforms: Architecture and Constraints (3 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Development – Native versus cross-platform development – Software design/architecture patterns for mobile applications API Design and Modeling, Principles and Software Architecture, Software Design Quality and Evaluation • Mobile platform constraints – Responsive user interface design Accessibility and Inclusive Design – Heterogeneity and mobility of devices – Differences in user experiences (e.g., between mobile and web-based applications) – Power and performance tradeoff • Specification and testing Software Development Practices, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices	• Develop a location-aware mobile application with data API integration [Create] • Build a sensor-driven mobile application capable of logging data on a remote server [Create] • Create a communication app incorporating telephony and instant messaging [Create]
Readings : [**Deitel2017**], [**Hellman2013**]	

Unit 6: Mobile Platforms: Capabilities and Asynchronous Computing (3 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Mobile computing affordances – Location-aware applications – Sensor-driven computing (e.g., gyroscope, accelerometer, health data from a watch) – Telephony and instant messaging – Augmented reality Immersion • Asynchronous computing Algorithms – Difference from traditional synchronous programming – Handling success via callbacks – Handling errors asynchronously – Testing asynchronous code and typical problems in testing	• Develop a location-aware mobile application with data API integration [Create] • Build a sensor-driven mobile application capable of logging data on a remote server [Create] • Create a communication app incorporating telephony and instant messaging [Create]
Readings : [**Deitel2017**], [**Hellman2013**]	

Unit 7: SPD-SEP/Interactive (10 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Ethical guidelines when using AI models to assist in journalism and content creation • Accountability for AI-generated outputs • Behavior among prompt programmers and AI developers • Trust with the public when using AI models	• Indicate a framework for accountability in AI model deployment, including clear documentation and attribution [Evaluate] • Discuss ethical codes of conduct and professional standards relevant to prompt programming and AI development [Debate] • Create communication plans and materials to educate the public about AI capabilities, limitations, and ethical safeguards [Create]
Readings : [**Duckett2014**], [**Deitel2017**]	

Unit 8: Interactive Data and Analytics Platforms (6 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Computational Notebook Platforms (e.g., Jupyter notebooks; Google Colab; R; SPSS; Observable) • Cloud SQL and Data Analysis Platforms (e.g., BigQuery) Query Construction – Apache Spark – Data Visualizations Visualization • Interactive Presentations Backed by Data • Design Tools Requiring Low-Latency Feedback Loops – Rendering tools – Graphic design tools	• Analyze large datasets interactively [Analyze] • Create a backing track for a musical performance, such as live coding [Create] • Create compelling computational notebooks that construct a narrative for a given journalistic goal/story [Create] • Implement interactive code that uses a dataset and generates exploratory graphics [Implement] • Contextualize the attributes of different data analysis styles, such as interactive vs engineered pipeline [Contextualize] • Write a program using a notebook computing platform (e.g., searching, sorting, or graph manipulation) [Write]
Readings : [**Flanagan2020**]	

7. WORKPLAN

7.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

7.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

7.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

8. EVALUATION SYSTEM

********* EVALUATION MISSING ********

9. BASIC BIBLIOGRAPHY