**Bayero University, Kano**

**Faculty of Computing**

**Department of Software Engineering**

**Proposed 30% addition to the CCMAS Course Structure**

**Level 100**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Title** | **Unit** | **Status** | **LH** | **PH** |
| BUK-COS 101 | Introduction to Application Packages | 2 | C | 30 | 45 |
| BUK-ICT 102 | Introduction to Information and Communication Technology | 2 | C | 30 | - |
| BUK-COS 103 | Introduction to Computer Programming | 3 | C | 30 | 45 |
|  | Total | **7** |  |  |  |

**Level 200**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Title** | **Unit** | **Status** | **LH** | **PH** |
| BUK-IFT 201 | Introduction to Web Technologies | 2 | E | 15 | 45 |
| BUK-INS 202 | Human Computer Interactions | 2 | C | 15 | 45 |
| BUK-SEN 203 | Software Engineering Process | 2 | E | 30 | - |
|  |  | 6 |  |  |  |

**Level 300**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Title** | **Unit** | **Status** | **LH** | **PH** |
| BUK-DTS 301 | Data Management I | 3 | C | 15 | 45 |
| BUK-ICT 302 | Data Communication Systems and Network | 3 | C | 30 | 45 |
| BUK-IFT 303 | Mobile Application Development | 2 | C | 15 | 45 |
| BUK-DTS 304 | Big Data Computing | 2 | C | 15 | 45 |
| BUK-MTH 305 | Introduction to Numerical Analysis | 2 | E | 15 | 45 |
|  | Total | **12** |  |  |  |

**Level 400**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Title** | **Unit** | **Status** | **LH** | **PH** | |
| BUK-SEN 401 | Software Engineering Professional Practice | 2 | C | 30 | | - |
| BUK-SEN 402 | Concepts of Programming Languages | 2 | C | 30 | | - |
| BUK-SEN 403 | Open Source Software Development and Applications | 2 | C | 15 | | 45 |
| BUK-SEN 404 | Special Topics in Software Engineering: | 2 | C | 15 | |  |
| BUK-SEN 405 | Software Engineering Economics | 2 | C | 30 | |  |
| BUK-CSC 406 | Artificial Intelligence | 2 | C | 15 | | 45 |
| BUK-CYB 407 | Cloud Computing | 2 | C | 15 | | 45 |
| BUK-IMT 408 | Operation Research | 2 | C | 30 | | - |
| BUK-SEN 409 | Game Design and Development | 2 | E | 15 | | 45 |
|  | Total | 18 |  |  | |  |
|  | | | | | | |
|  | Total Core | **35** |  |  | |  |
|  | Total Elective | **08** |  |  | |  |
|  | Total Core and Elective | **43** |  |  | |  |

**Bayero University, Kano**

**Faculty of Computing**

**Department of Software Engineering**

**BSc Software Engineering**

**BUK-COS 101 Computer Application Packages. (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

This course is aimed at beginning to intermediate computer users. It teaches a range of computer skills from the basics of using Windows, to basic internet literacy, to creating projects using Microsoft Office. Assignments show step-by-step visuals to help student’s complete projects, and include integration across Microsoft Word and Excel. This course will apply learning sciences to engage students and better support the learning process. The material is aimed to deliver an active learning experience. They include text, images, videos, assessments, directed feedback, and practice questions that invite students to apply their knowledge, improve their understanding, and perform better.

**Objectives**

The objectives of the course are to:

1. Describe navigation and perform common tasks in Word, such as opening, viewing, editing, saving, and printing documents, and configuring the application.
2. Discuss formatting text, paragraphs and repetitive operations efficiently using tools such as Find and Replace, Format Painter, and Styles.
3. Explore lists by sorting, renumbering, and customizing list styles, create and format tables.
4. Outline graphic objects into a document, including symbols, special characters, illustrations, pictures, and clip art, format the overall appearance of a page through page borders and colors, watermarks, headers and footers, and page layout.
5. Explain working with Multiple Worksheets and performing computations in Worksheets.

**Learning Outcomes**

The students shall be able to:

1. Select the most appropriate software to use to complete a task
2. Identify the key features of a word processor and spread sheet application
3. Apply the key features of a word processor to format a text, paragraph and document
4. Evaluate formatting techniques to understand why we format documents
5. Develop and apply fundamental spreadsheet skills. And demonstrate proficiency in using moderately complex spreadsheet tools such as tables and chart

**Course Contents**

Introduction to Computer Application; what are computer application packages, types of computer application packages Overview of popular computer application packages. Microsoft Office Suite Microsoft Word; Creating, editing, and formatting documents, working with tables, graphics, and charts, collaborating on documents. Microsoft Excel; Creating, editing, and formatting spreadsheets, working with formulas and functions, Creating charts and graphs Microsoft PowerPoint; Creating, editing, and formatting presentations, Adding multimedia elements, Delivering presentation.

**Minimum Academic Standards**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-ICT 102 Introduction to Information and Communication Technology. (2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

The Fundamentals of Information Technology course is designed to provide students with an understanding of the basic principles of computing and information technology. The course covers a wide range of topics, including computer hardware and software, computer networks and the internet, database management systems, programming languages, and cybersecurity.

Throughout the course, students will have the opportunity to apply what they have learned through hands-on exercises and projects. They will also be introduced to tools and technologies commonly used in the field of information technology, such as software development environments, network monitoring tools, and database management systems. Overall, the Fundamentals of Information Technology course provides students with a solid foundation in the principles of computing and information technology, preparing them for further study in the field or for entry-level positions in the IT industry.

**Objectives**

The objectives of the course are to:

1. Develop a basic understanding of the history and evolution of information technology and its impact on society.
2. Apply knowledge of computer hardware and software components, including the ability to identify, install, and configure computer hardware.
3. Apply knowledge of emerging trends in information technology, such as cloud computing, artificial intelligence, and the internet of things.
4. Explain the ethical and social implications of information technology and its impact on society.
5. Discuss the impact of ICT on society, including ethical and legal issues.

**Learning Outcomes**

The students shall be able to:

1. Identify components of the computer and know-how the components communicate;
2. Apply the concept of data transfer and memory types and management and be comfortable with graphics processing;
3. Identify and be able to use different communication ports and know the software types;
4. State different computer network topologies and their sizes; and
5. Operate office applications and the internet.

**Course Content**

Basic principle of computers. Computer "backbone". Data transmission. Random Access Memory. Permanent Memory. Graphic processing. Communication Ports. Input and Output Devices. Software types. Accessibility options. Computer types. Portable digital devices. Network Types. Internet. Instant messaging. Voice over Internet Protocol. Really Simple Syndication. Network communication. Internet data transfer. Data rate units. Internet access. Virtual (online) communities. Computer in the workplace. Telecommuting (telework).

**Minimum Academic Standards**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University, Kano.**

**Faculty of Computing**

**Department of Software Engineering**

**BSc. Software Engineering**

**BUK-COS-103 Introduction to computer programming. 3 Unit, Core, LH=30, PH=45**

**Senate approved relevance to mission and strategic goals of the university**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines**.**

**Overview**

In this course, students will learn how Python works and its place in the world of programming languages; to work with and manipulate strings; to perform math operations; to work with Python sequences; to collect user input and output results; flow control processing; to write to, and read from, files; to write functions; to handle exception; and work with dates and times.

**Objectives**

The objectives of the course are to:

1. Explain why Python is a useful scripting language for developers

2. Develop and program Python applications.

3. Discuss how to use lists, tuples, and dictionaries in Python programs and identify Python object types.

4. Explain how to use indexing and slicing to access data in Python programs, write loops and decision statements in Python.

5. Apply functions and pass arguments in Python, build and package Python modules for reusability and read and write files and design object‐oriented programs with Python classes

**Learning outcome**

The students will be able to:

1. Create basic programs using fundamental programming constructs like variables, conditional logic, looping, and functions

2. Create indexing and slicing to access data in Python programs

3. Create user input to create fun and interactive programs

4. Create simple games with images, animations, and audio using our custom beginner-friendly programming library, Wizardlib

5. Create and manipulate files

**Course Content**

Vital Python – Math, Strings, Conditionals, and Loops. Vital Python. Numbers: Operations, Types, and Variables. To Open a Jupyter Notebook. Python as a Calculator. Standard Math Operations. Basic Math Operations. Order of Operations. Spacing in Python. Number Types: Integers and Floats. Complex Number Types. Errors in Python. Variables. Variable Assignment. Changing Types. Reassigning Variables in Terms of Themselves. Variable Names. Multiple Variables. Comments. Docstrings. Theorem in Python. Strings: Concatenation, Methods, and input(). String Syntax. Escape Sequences with Quotes. Multi-Line Strings. The print() Function. String Operations and Concatenation. String Interpolation. Comma Separators. Format. The len() Function. String Methods. Casting. The input() Function. String Indexing and Slicing. Indexing. Slicing Strings and Their Methods. Booleans and Conditionals. Booleans. Logical Operators. Comparison Operators. Comparing Strings. Conditionals. The if Syntax. Indentation. if else. The elif Statement. Loops. The while Loops. An Infinite Loop. break. Programs. The for Loop. The continue Keyword. Python Structures. The Power of Lists. List Methods. Accessing an Item from a List. Adding an Item to a List. Dictionary Keys and Values. a List and a Dictionary. Zipping and Unzipping Dictionaries Using zip(). Dictionary Methods. Tuples. A Survey of Sets. Set Operations. Choosing Types. Executing Python – Programs. Algorithms, and Functions Introduction. Python Scripts and Modules.

**Minimum Academic standards requirements**

Computer Science  programme’s NUC-MAS requirement facilities

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-IFT-201 Introduction to Web Technologies. (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

The Web Application Development course is designed to teach students how to develop web-based applications using modern web technologies. Throughout the course, students will learn how to design and develop web applications that are both functional and user-friendly. Students will be introduced to fundamental web development concepts, including HTML, CSS, and JavaScript. They will learn how to use these technologies to structure web pages, style them, and add interactivity and functionality to web applications.

The course will also cover popular web development frameworks such as React, Angular, and Vue.js, which students will use to build dynamic, scalable web applications. In addition, students will learn about server-side programming languages like Node.js and Ruby on Rails and how to use them to build robust backend systems that power web applications.

**Objectives**

The objectives of the course are to:

1. Provide students with a solid understanding of web development concepts and technologies, including HTML, CSS, JavaScript, and web frameworks.
2. Design and develop responsive, user-friendly web applications using modern web technologies.
3. Create server-side programming languages like Node.js and Ruby on Rails, and how to use them to build robust backend systems that power web applications.
4. Create and utilize database technologies to store and manage data within web applications.
5. Explain to students web application security and performance best practices, including techniques for securing web applications against common vulnerabilities and optimizing web application performance.

**Learning Outcomes**

Students shall be able to;

1. Design and implement simple client-side and server-side web applications;
2. Demonstrate hands-on skills in PHP and Python programming uses open-source software;
3. Distinguish web programming with general-purpose programming; and
4. Develop a fully functioning website and deploy it on a web server.
5. Design web application with sound security.

**Course Content**

Introduction to framework-based web development using a contemporary language like PHP and ASP.net. Principles of web pages (dynamic and static) and website design. The tool used in web development of Client-side and server-side languagesCreation of interactive, dynamic websites using a common web architecture and object-based database accessDesign, implementation, and testing of web-based applications including related software, databases, interfaces, and digital mediaStandard object models, and the use of server-side programmes for database and file access; testing, software quality assurance; and the process of publishing Web sitesHands-on PHP and Python programme using open-source software (Apache, PHP, Python, JavaScript, and MySQL)Programming for web development includes control structures, objects, functions, and the use of composite data typesDeploying dynamic content using JavaScriptDesigning and developing dynamic web pages and creating, validating, transforming, and formatting data using PHP.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN-203 Software Engineering Process**. **(2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

A software engineering process course typically covers the various phases involved in the development of software products. It is designed to give students an in-depth understanding of the software development life cycle and the methods and techniques used to create high-quality software.

By the end of the course, students should have a comprehensive understanding of the software development process and the tools and techniques used to create high-quality software products. They should also be able to apply these concepts to real-world software development projects

**Objectives**

The objectives of the course are to:

1. Prepare students with an understanding of software engineering process, its principles, and practices.
2. Prepare students with the knowledge and skills required to design, implement and maintain software systems.
3. Apply software engineering principles and practices to real-world problems.
4. Develop an appreciation for the importance of software engineering in modern society.
5. Prepare students for a career in software engineering.

**Learning Outcomes**

Students shall be able to;

1. Identify the various phases of the software engineering process and explain the purpose of each phase.
2. Apply software engineering principles to design and develop software systems that meet user requirements.
3. Evaluate and select appropriate software engineering techniques and tools for a given problem.
4. Design, implement, and test software systems.
5. Explain to stakeholders software engineering issues, including project requirements, progress, and outcomes.

**Course Content**

Software process definition – software process management and infrastructure, Software life cycles – categories of software processes, software life cycle models, software process adaptation, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK- INS 202 Human-Computer Interaction. (2 Unit; Core; LH=15, PH=45)**

**Senate approved relevance to mission and strategic goals of the university**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Human-Computer Interaction (HCI) is a field that explores the design, development, and evaluation of interactive computer systems for human use. The course provides an overview of HCI concepts, principles, and techniques that enable the design and development of effective interactive systems.

The course covers the basics of human perception, cognition, and behavior, as well as the ways in which people interact with computer systems. Students will learn about the principles of usability, user-centered design, and user experience design, and the role of feedback, interactivity, and feedback in designing effective interactive systems.The course also covers a range of design and evaluation methods, such as user research, prototyping, and usability testing. Students will learn how to identify user needs, design interfaces that meet those needs, and evaluate the usability and effectiveness of interactive systems.

Finally, the course explores emerging trends in HCI, such as mobile computing, social computing, and ubiquitous computing. Students will learn how to design and develop interactive systems that are adapted to different contexts and platforms, and how to evaluate the effectiveness of such systems.

By the end of the course, students should have a solid understanding of the fundamental concepts of HCI, and be able to apply these concepts to the design and development of effective interactive systems. Students will also gain hands-on experience in designing, prototyping, and evaluating interactive systems, and be prepared for further study or careers in HCI, design, or user experience.

**Objectives**

The objectives of the course are to:

1. Explain the fundamentals of human-computer interaction (HCI) and its importance in design.

2. Develop skills in analyzing, designing, and evaluating user interfaces for digital systems and gain knowledge of various design principles, guidelines, and best practices for creating effective user interfaces.

3. Explore different approaches to user research and testing to gather user feedback and improve design, emerging technologies and their impact on HCI, such as virtual and augmented reality, conversational interfaces, and machine learning.

4. Develop critical thinking skills to evaluate the ethical and social implications of technology design and use.

5. Explore and analyze case studies and real-world examples of successful and unsuccessful HCI design.

**Learning Outcomes**

Students shall be able to;

1. Discuss the foundations and concept of the human-computer interface;

2. Explain Understanding of principles of human-computer interface;

3. Explain the design and development of the human-computer interface; and

4. Explain the importance of user feedback.

**Course content**

Foundations of HCI. The concept underlying the design of HCI. Principles of GUI. GUI toolkits. System design methods. User conceptual models and interface metaphors. Human cognitive and physical ergonomics. Human-centred software evaluation and development. GUI design and programming.

**Minimum Academic standards requirements**

Computer Science  programme’s NUC-MAS requirement facilities

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-DTS-301 Data Management I**. **(2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Database Management System (DBMS) course typically covers the principles, design, implementation, and management of databases. The course aims to equip students with the necessary skills and knowledge to design and manage efficient and reliable databases that can store, organize, and retrieve data for various applications.

A database management course typically covers the principles and techniques used to design, implement, and manage a database system. The course may cover both relational and non-relational databases.

**Objectives**

The objectives of the course are to:

1. Provide a solid understanding of the basic concepts and principles of data management.
2. Explore data management tools and technologies.
3. Develop skills in data analysis and reporting.
4. Apply data security and privacy concepts.
5. Develop skills in project management.

**Learning Outcomes**

Students shall be able to;

1. Describe the components of a database system and give examples of their use.
2. Describe the differences between relational and semi-structured data models.
3. Explain and demonstrate the concepts of entity integrity constraint and referential integrity  
   constraint.
4. Apply queries, query optimizations and functional dependencies in relational databases.
5. Describe database security and integrity issues and their importance in database design.

**Course Content**

Information Management Concepts. Information storage & retrieval. Information management applications. Information capture and representation. Analysis and indexing -search, retrieval, information privacy. Integrity and security. Scalability, Efficiency and Effectiveness. Introduction to database systems. Components of database systems. DBMS  
functions. Database architecture and data independence. Database query language. Conceptual models. Relational data models. Semi-structured data models. Relational theory and languages. Database Design. Database security and integrity. Introduction to query processing and optimisation. Introduction to concurrency and recovery

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-ICT-302 Data Communication Systems and Network I. (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Data networking and communication is a course that covers the fundamentals of computer networks, including how data is transmitted and received over networks, as well as the different types of network topologies and protocols used in data communication.

Data communication course is designed to provide students with a fundamental understanding of data communication and networking concepts such as networking architecture, transmission media, and protocol and network management

**Objectives**

The objectives of the course are to:

1. Explain the fundamental concepts of data communication, including how data is transmitted and received, different types of communication channels, and the different types of transmission protocols.
2. Discuss networking technologies such as LAN, WAN, WLAN, VPN, and TCP/IP. Students may learn how these technologies work, their advantages and disadvantages, and how to set them up.
3. Explain the various threats to network security and how to protect networks from these threats. Students may learn about encryption, firewalls, access controls, and other security measures.
4. Explore how to manage networks, install and configure network components, troubleshoot network problems, and perform network maintenance.
5. Apply knowledge to real-world situations to gain hands-on experience with networking technologies and solve networking problems.

**Learning Outcomes**

Students shall be able to:

1. Explain data transmission over layered networks;
2. List and explain common internet technologies and protocols; and
3. Explain network operating system
4. Design and configure networks:
5. Identify network problems, perform tests, analyze results, and take corrective action.

**Course Content**

Types and sources of data. Simple communications network. Transmission definitions, one  
way transmission, half duplex transmission, transmission codes, transmission modes, parallel  
transmission, serial transmission, bit synchronisation, character synchronisation, synchronous and asynchronous transmission. Introduction to network protocol. Seven Layer ISO-OSI standard protocols and  
network architecture. Transport protocols, session services protocols, and other protocols.  
Institute of Electrical and Electronics Engineering 802 standards. Error control and Data  
Compression: Forward Error Control; error detection methods; parity checking; linear block  
codes, cyclic redundancy checking; feedback error control, data compression, Huffman coding.Local Area Networks: medium access control techniques; fibre distributed data interface, and metropolitan area network. Peer-to-peer, Client Server. Client-Server Requirements: GUI design standards, interface and platform independence, transaction processing, backup, and recovery mechanisms. Features and benefits of major recovery  
mechanisms. Network OS: INTERNET: Definition, architecture, services, internet addressing. Internet protocol, IPv4,  
IPv6

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-IFT-303 Mobile Application Development. (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other **Overview**

The Mobile Application Development course is designed to teach students how to design and build mobile applications for different platforms, such as Android and iOS. In this course, students will learn about the different mobile operating systems, software development kits (SDKs), and mobile app development frameworks that are used in mobile application development.

The Mobile Application Development course is designed to prepare students for careers in mobile application development or related fields by providing them with the skills and knowledge necessary to design and build high-quality mobile applications that meet the needs of users and organizations. Throughout the course, students will gain a solid understanding of key mobile application development concepts such as mobile UI design, navigation, data storage, and integration with other mobile devices. Additionally, they will be introduced to different mobile application development platforms, such as Android and iOS, and will learn how to develop applications for these platforms using appropriate tools and technologies.

**Objectives**

The objectives of the course are to:

1. Discuss concepts and technologies used in mobile application development, including mobile operating systems, software development kits (SDKs), and mobile app development frameworks.
2. Explore how to apply debugging, testing, and version control techniques to mobile application development projects.
3. Discuss different mobile application development platforms, such as Android and iOS, and teach them how to develop applications for these platforms using appropriate tools and technologies.
4. Discuss mobile applications for different mobile devices, such as smartphones and tablets, and for different mobile networks and connectivity conditions.
5. Prepare students for careers in mobile application development or related fields by providing them with the skills and knowledge necessary to design and build high-quality mobile applications that meet the needs of users and organizations.

**Learning Outcomes**

Students shall be able to:

1. Identify the basic knowledge on mobile application environment and technology;
2. Explain the concepts and processes of mobile application development;
3. Discuss design and development issues specific to mobile applications;
4. Design and develop mobile applications, using development tools and environments;
5. Evaluate the performance of a mobile application and give its result; and appreciate perspectives of mobile applications and their impact.

**Course Content**

Introduction to developing mobile applications. Mobile operating systems capabilities, application architecture, and major components, such as activities, services, broadcast receivers, etc. Development of interactive applications using widget libraries, web-based services. Basic concepts of 2D graphics and animation. An SQL database engine, and multithreading. Multiplatform mobile application development. Mobile application basics and features; Android application basics, UI design Data storage; networking application design. Advanced application design (sensors, camera, GPS, Audio, etc.), graphics and games, web-based hybrid application design. Design and implement a simple mobile application for a given mobile platform. Metrics and methods to evaluate the performance of mobile applications. Mobile application perspectives and impact.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-DTS 304 Big Data Computing. (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

The Big Data Computing course is designed to teach students about the challenges and opportunities presented by the processing and analysis of large-scale data sets, also known as big data. The course covers a range of topics related to big data, including data storage, distributed computing, data management, and analysis.

Throughout the course, students will gain an understanding of the technologies and tools used in big data computing, such as Hadoop, MapReduce, and Spark. They will also learn about data management techniques for big data, including data warehousing and data lakes, and the importance of data quality and integrity in big data systems. The course will also cover data analysis techniques for big data, such as machine learning, data mining, and predictive analytics.

**Objectives**

The objectives of the course are to:

1. Explain the concepts and challenges of big data computing, including data storage, distributed computing, and data management.
2. Explore data management techniques for big data, including data warehousing and data lakes, and the importance of data quality and integrity in big data systems.
3. Explain to students about data analysis techniques for big data, such as machine learning, data mining, and predictive analytics.
4. Provide students with hands-on experience in processing and analyzing large-scale data sets using big data technologies and tools.
5. Prepare students for careers in big data computing or related fields by providing them with the skills and knowledge necessary to manage, process, and analyze large-scale data sets.

**Learning Outcomes**

Students shall be able to;

1. Identify Big Data;
2. Identify some of the foundational tools, systems, and platforms that feature in working with Big Data across several domains;
3. Install Big Data working tools on a computer; and
4. Analyze Big Data contents.
5. Differentiate types of data storage and management systems used in big data computing, including distributed file systems like Hadoop Distributed File System (HDFS) and NoSQL databases

**Course Content**

Installation: Cloudera VM, Jupyter server. Big data retrieval and relational querying: Postgres databases, NoSQL data, MongoDB, Aerospike, and Pandas for data aggregation and working with data frames. Big Data Integration: Splunk and Datameer. Big Data Processing: Apache Spark, Hadoop, Spark Core (Spark MLlib and GraphX. Big Data Applications (Graph Processing). Big Data Streaming Platforms for Fast Data. **Lab Work:**Analysing Twitter Data using Spark and MongoDB. Learn Big Data analytics skills Practical procedure for the crafting of an enterprise-scale cost-efficient. Big Data and machine learning solution to uncover insights and value from data. Hands-on exposure to Hadoop and Spark (or any of the BD tools), build machine learning dashboards using R and R Shiny, and create web-based apps using NoSQL databases.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-MTH 305 Introduction to Numerical Analysis. (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Introduction to Numerical Analysis is a course designed to introduce students to the fundamental principles and techniques of numerical computation. The course covers a range of topics, including numerical methods for solving equations, numerical differentiation and integration, and numerical solutions of ordinary and partial differential equations.

The course typically starts with an overview of mathematical modeling and how it relates to numerical computation. Then, students will learn about numerical methods for solving equations, including the bisection method, Newton's method, and the secant method. The course will also cover error analysis and convergence rates for these methods. Introduction to Numerical Analysis provides students with the tools and knowledge needed to apply numerical methods to solve a variety of problems in science, engineering, and mathematics. The course emphasizes both theory and practice, with students gaining hands-on experience through programming assignments and projects.

**Objectives**

The objectives of the course are to:

1. Develop an understanding of numerical methods and their applications.
2. Develop the ability to analyze and solve mathematical problems using numerical techniques.
3. Develop skills Gain in programming and using numerical software tools and interpret numerical results and communicate them effectively.
4. Acquire a foundation in mathematical analysis and linear algebra to support the study of numerical methods.
5. Develop the ability to apply numerical methods to real-world problems in computer science, engineering, physics, economics, and other fields.

**Learning Outcomes**

Students shall be able to;

1. solve some numerical solution of algebraic and transcendental equations and describe curve fitting;
2. discuss error analysis;
3. calculate interpolation and approximation;
4. solve some numerical differentiation and numerical integration problems; and
5. solve some numerical problems in ordinary Differential equations with initial value problems;

**Course Content**

Solution of algebraic and transcendental equations. Curve fitting. Error analysis. Interpolation and approximation. Zeros of non-linear equations ‘in one variable’. Systems of linear equations. Numerical differentiation and integration. Initial value problems in ordinary differential equation

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN-401 Software Engineering Professional Practice (2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Software Engineering Professional Practice is a course that focuses on the application of software engineering principles, practices, and standards in a professional setting. The course covers a range of topics, including software development life cycle models, software project management, software requirements engineering, software testing, software quality assurance, and software maintenance.

Throughout the course, students will learn about the importance of professional ethics and the legal and social implications of software engineering. They will also develop skills in communication, teamwork, and project management, which are essential for success in the software engineering industry.

**Objectives**

The objectives of the course are to:

1. Provide an understanding of the software development lifecycle and its various phases.
2. Discuss the principles and best practices of software engineering, including software design, coding standards, and testing techniques.
3. Develop skills in software development methodologies, such as agile and waterfall.
4. Discuss the principles and techniques of software project management, including requirements gathering, scheduling, and risk management.
5. Apply software engineering concepts and techniques to real-world software development projects.

**Learning Outcomes**

Students shall be able to;

1. Develop an understanding of the software development lifecycle, including the various phases of software development and their importance.
2. Apply software engineering principles and best practices, including software design, coding standards, and testing techniques, to develop high-quality software.
3. Apply software development methodologies, such as agile and waterfall, to develop software in a collaborative and iterative manner.
4. Develop skills in software project management, including requirements gathering, scheduling, and risk management, to effectively manage software projects.
5. Apply software engineering concepts and techniques to real-world software development projects, demonstrating the ability to develop high-quality software that meets the needs of users and stakeholders.

**Course Content**

Software development life cycle: various phases involved in software development, such as planning, requirements analysis, design, coding, testing, and maintenance. Software project management: project planning, scheduling, estimation, risk management, and quality assurance. Software testing: unit testing, integration testing, system testing, and acceptance testing. Software quality assurance: code reviews, static analysis, and automated testing. Software configuration management: version control, release management, and change management.Software documentation: requirements documents, design documents, user manuals, and technical specifications.Software ethics: privacy, security, and intellectual property.Professionalism: effective communication, teamwork, and continuous learning

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN 402 Concepts of Programming Languages (2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Concept of programming languages course is designed to introduce students to the fundamentals of computer programming and the various programming languages used in software development. The course covers topics such as syntax, data types, variables, control structures, functions, arrays, object-oriented programming, and error handling.

The course typically starts by introducing the basic concepts of programming, including algorithms, data structures, and programming paradigms. Then, it delves into the syntax and semantics of programming languages such as C, Python, Java, and others. Students learn how to write simple programs and gradually work their way up to more complex applications.

**Objectives**

The objectives of the course are to:

1. Discuss the fundamental concepts of programming languages, such as syntax, semantics, and programming paradigms.
2. Explain the importance of selecting the appropriate programming language for a given problem and to make informed decisions about language selection.
3. Explain how to write correct and efficient code using the chosen programming language.
4. Discuss advanced topics in programming language design and implementation, such as compiler construction and language theory.
5. Prepare students for further study and research in programming languages and related fields.

**Learning Outcomes**

Students shall be able to;

1. Describe and compare the syntax and semantics of different programming languages.
2. Create code in at least one programming language, and understand the differences in programming paradigms.
3. Evaluate the strengths and weaknesses of different programming languages and choose the most appropriate language for a given problem.
4. Design and implement small programming languages and interpreters, and understand the basics of compiler construction.
5. Apply knowledge to their own programming practice.

**Course Content**

Introduction to Programming Languages**;** what is a programming language? Types of programming languages (procedural, functional, object-oriented, etc.)Syntax and semantics of programming language. History of Programming Languages**;** Early programming languages, Evolution of programming languages, Influence of hardware and software developments on programming languages**.** Language Translation**;** Compilers, interpreters, and virtual machines**,** Lexical analysis and parsing**,** intermediate representation. Data Types and Variables**;** Basic data types, User-defined data types**,** Variables and constants**.** Control Structures**;** Sequence, selection, and iteration**,** Conditionals and loops**,** Exception handling**.** Functions and Procedures**;** Definition and calling of functions**,** Parameter passing mechanisms**,** Scope and lifetime of variables. Object-Oriented Programming**;** Classes and objects**,** Inheritance and polymorphism**,** Interfaces and abstract classes. Functional Programming**;** Lambda calculus**,** Higher-order functions**,** Immutable data structures**.** Language Design and Implementation**;** Language syntax and semantics**,** Language features and trade-offs**,** Language implementation and optimization**.** Current Trends in Programming Languages**;** New programming languages (Rust, Kotlin, Swift, etc.)**,** Domain-specific languages (DSLs)**,** Web programming languages (JavaScript, TypeScript, etc.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN 403 Open Source Software Development and** **Applications (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Open source software development course is designed to teach creating software that allows anyone to view, modify, and distribute the source code. It is a popular approach for creating high-quality software that is free and openly available to everyone.

The course is designed to teach how to develop software using open source tools and technologies. The goal of open source development is to create high-quality innovative software that is freely available to everyone.

**Objectives**

The objectives of the course are to:

1. Explain the principles of open source software development and its importance in the industry.
2. Acquire the skills and knowledge necessary to contribute to open source software projects.
3. Discuss how to use tools and platforms commonly used in open source software development, such as Git and GitHub.
4. Explore the different types of open source licenses and their implications on software development.
5. Create real-world open source software project.

**Learning Outcomes**

Students shall be able to;

1. Develop a thorough understanding of the principles of open source software development, including its history and benefits.
2. Acquire practical skills in using open source software development tools, such as version control systems and bug trackers.
3. Explain the importance of communication and collaboration within the open source community, including how to interact with other developers and contribute effectively to open source projects.
4. Explain the implications of different types of open source licenses on software development, including how to choose a license and how to comply with license requirements.
5. Develop a strong sense of responsibility and commitment to contributing to the open source community, including how to create and maintain high-quality open source software.

**Course Content**

Introduction to open source software development: what open source software development is and why it is important.Open source licensing: different types of open source licenses and their implications. Tools and technologies: This will cover the various tools and technologies used in open source development such as version control systems, issue trackers, and collaborative platforms.Building and contributing to open source projects: how to find and contribute to existing open source projects as well as how to start your own. Best practices for open source development: code reviews, testing, documentation, and communication. Community management: how to manage an open source community and how to handle conflicts that may arise.Open source business models: business models that can be used for open source software development such as support and services, dual licensing, and open core.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN 404 Special Topics in Software Engineering (2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

This course will cover advanced topics in software engineering beyond the basics of software development. The course will focus on current research and emerging technologies, with an emphasis on practical applications.

Recent topics and developments in software engineering are expected to be introduced from year to year. Apart from seminars to be delivered by lecturers or guests, students are expected to do substantial readings on their own

**Objectives**

The objectives of the course are to:

1. Develop advanced skills and knowledge in a particular area of software engineering, such as software testing or software security.
2. Explore emerging trends and technologies in software engineering, such as machine learning or block chain.
3. Develop practical skills in software engineering project management and leadership.
4. Develop research skills in software engineering.
5. Develop skills in software engineering entrepreneurship and innovation.

**Learning Outcomes**

Students shall be able to;

1. Apply advanced software engineering techniques and tools to solve complex problems in the chosen area, demonstrate an understanding of the key concepts and principles of the area, and evaluate and critique current research and practices in the area.
2. Evaluate the potential benefits and limitations of these emerging trends and technologies in software engineering, identify and explain key concepts and techniques, and assess the suitability of these approaches for specific software engineering problems
3. Create and manage software engineering projects, including estimating project costs and schedules, defining project scope and objectives, identifying and managing risks, and leading project teams. Students will also be able to effectively communicate project status and outcomes to stakeholders.
4. Design and conduct research in software engineering, including selecting appropriate research methods and tools, gathering and analyzing data, and interpreting and reporting research results. Students will also be able to critically evaluate research literature and identify areas for future research.
5. Identify and evaluate opportunities for software engineering innovation, develop and pitch new software engineering products or services, and create and execute business plans for software engineering startups. Students will also be able to identify and manage the risks associated with software engineering entrepreneurship.

**Course Content**

Introduction to Advanced Software Engineering; Overview of software engineering principles and methodologies**,** emerging trends in software development**,** Introduction to advanced software engineering concepts**.** Agile Software Development**;** Principles of agile software development**,** agile methodologies and practices**,** Managing agile projects**.** Cloud Computing and Micro services**;** Cloud computing concepts and architectures**,** Micro services and service-oriented architectures**,** Designing and implementing micro services**.** Emerging Technologies in Software Engineering; Blockchain and distributed ledger technology**,** Artificial intelligence and machine learning in software engineering**,** Quantum computing and its potential impact on software development**.** Research papers and journal that are published in special topics areas**.** Open discussion on current and research and industry trends related to the special topic area. Case Studies and Project Presentations.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN 405 Software Engineering Economics (2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

This course provides an overview of the economic principles and techniques that are essential to the management of software development projects. The course focuses on the economics of software engineering which include cost and benefit analysis of software development projects.

Throughout the course Students will learn how to apply these principles to make informed decisions regarding software development processes, project planning, cost estimation, risk management, and project control.

**Objectives**

The objectives of the course are to:

1. Discuss estimate the cost of developing software projects accurately.
2. Explain allocating resources optimally, including people, time, and money.
3. Identify and manage risks that could affect the success of a project.
4. Create tasks and activities in a way that maximizes efficiency and minimizes delays.
5. Evaluate and Measure the performance of software engineering projects.

**Learning Outcomes**

Students shall be able to;

1. Apply basic economic principles, such as supply and demand, to software engineering problems.
2. Estimate the cost of developing software projects using different techniques and models, including bottom-up and top-down estimation methods.
3. Identify and manage risks that could affect the success of software engineering projects, including technical, project, and business risks.
4. Create and manage project schedules using techniques such as Gantt charts, network diagrams, and critical path analysis.
5. Measure and evaluate the performance of software engineering projects, including identifying project success criteria and using metrics to evaluate project performance.

**Course Content**

Introduction**;** Overview of software engineering economics**,** Importance of software engineering economics in software development**,** Basic concepts and terminologies**.** Software Development Process Models**;** Waterfall Model**,** Agile Development**,** Hybrid models**,** Comparison of models in terms of cost, time and quality**.** Software Cost Estimation**;** Cost drivers**,** Cost estimation techniques (e.g., COCOMO, Function Points)**,** Uncertainty and risk management**.** Comparison of techniques and their limitations**.** Software metrics**.** Software Economics and Decision Making**;** Time value of money**,** Return on investment (ROI)**,**Net present value (NPV) and discounted cash flow analysis**,** Cost-benefit analysis**,** Software project selection

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-CSC 406 Artificial Intelligence (2 Unit; Core; LH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Artificial intelligence course is designed to introduce students to the field of AI including its history, application and techniques. The course is typically designed to provide students with a comprehensive understanding of the principles and techniques used in AI as well as their application in various domains. The course covers a range of topics related to machine learning, natural language processing and computer vision.

In this course, students will learn about the fundamentals of AI, its various applications, and the techniques used to develop intelligence. Throughout the course students will work on projects related to AI using software tools such python

**Objectives**

The objectives of the course are to:

1. Provide an understanding of the basic concepts and principles of AI, including machine learning, neural networks, natural language processing, and robotics.
2. Introduce students to the various tools and techniques used in AI development, such as Python, TensorFlow, and PyTorch.
3. Enable students to apply AI techniques to real-world problems and develop AI systems that can learn from data and make predictions.
4. Familiarize students with the ethical and societal implications of AI, including issues related to privacy, bias, and fairness.
5. Prepare students for careers in AI research, development, and implementation and to provide a foundation for further study in specialized areas of AI, such as computer vision, speech recognition, and autonomous systems.

**Learning Outcomes**

Students shall be able to;

1. explain AI fundamentals, concepts, goals, types, techniques, branches, applications, AI  
   technology and tools;
2. discuss intelligent agents, their performance, examples, faculties, environment and  
   architectures, and determine the characteristics of a given problem that an intelligent  
   system must solve;
3. differentiate between the concepts of optimal reasoning/behaviour and human-like  
   reasoning/behaviour;
4. describe the role of heuristics and the trade-offs among completeness, optimality, time  
   complexity, and space complexity;
5. analyse the types of search and their applications in AI and describe the problem of  
   combinatorial explosion of search space and its consequences.

**Course Content**

Introduction to Artificial Intelligence: overview of what AI is, its history, and the various types of AI. Problem Solving using AI: various AI techniques to solve problems such as search algorithms, optimization algorithms, and constraint satisfaction. Machine Learning: basic concepts of machine learning, including supervised and unsupervised learning, decision trees, and neural networks. Natural Language Processing: parsing, semantic analysis, and sentiment analysis. Computer Vision: basics of computer vision, including image processing, feature extraction, and object recognition.Robotics: basics of robotics, including kinematics, control systems, and sensors. Ethics and Future of AI: discuss the ethical considerations around AI and its impact on society.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-CYB 407 Cloud Computing and Security (2 Unit; Core; LH=15, PH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Cloud Computing is a technology that allows you to use the resources of a large number of computers connected through a real-time communication network. This course defines Cloud Computing and establishes a strong working knowledge of the concepts and technologies needed to work effectively with the cloud. The course allows students to understand what cloud computing is and how it works. It describes the benefits of cloud computing along with its potential drawbacks.

The course covers technologies required to build classic (traditional), virtualized, and cloud data center environments. These technologies include compute, storage, networking, desktop and application virtualization.

**Objectives**

The objectives of the course are to:

1. Explain the basic concepts and terminology of cloud computing;
2. Explore areas of cloud technologies;
3. Discuss with the concept of cloud computing in relation to business activities;
4. Discuss security Studies, scaling, deployment, backup, in the context of cloud infrastructure;
5. Develop skills and techniques in cloud programming;

**Learning Outcomes**

Students shall be able to;

1. Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
2. Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Explain the core issues of cloud computing such as security, privacy, and interoperability.
4. Choose the appropriate technologies, algorithms, and approaches for the related issues.
5. Identify problems, and explain, analyze, and evaluate various cloud computing solutions.

**Course Content**

Introduction to cloud computing. Objectives, challenges, application domains, advantages. Computational and storage cloud architectures; Service level agreements, service lifecycle management. Elasticity and scalability techniques; Information, account and billing management. Cloud service model, service provisioning and access models. Cloud Service Models: Software as a Service layer; Platform as a Service layer; Infrastructure as a Service layer. Virtualization and resource management. Distributed object storage clouds. Data storage and retrieval based on content. Computational tasks execution in storage clouds. Quality of service approaches; Requirements and parameters classification, Monitoring and control mechanisms, Quality of service guarantees Security in the Cloud: Cloud threats; Threat Mitigation, Cloud and Security Risk.

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-IMT 408 Operation Research (2 Unit; Core; LH=15, PH=30)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

Operation research course is designed to introduce student to mathematical and quantitative methods are used to optimize complex systems and make informed decisions. This course provides an introduction to Operations Research, a field of study that uses mathematical and analytical methods to help organizations make better decisions.

The course covers a range of topics, including linear programming, network analysis, queuing theory, decision analysis, and simulation modeling. Emphasis is placed on the use of modeling techniques and optimization software to solve practical problems.

**Objectives**

The objectives of the course are to:

1. Develop a solid understanding of the fundamental principles and concepts of Operations Research.
2. Develop proficiency in formulating and solving linear programming problems, including optimization of resource allocation, production planning, and transportation problems.
3. Explain network analysis techniques for solving problems such as shortest path, maximum flow, and project scheduling.
4. Discuss Gain the role of Operations Research in a wide range of industries, including manufacturing, transportation, healthcare, and finance.
5. Explore the ethical considerations involved in Operations Research and its impact on society

**Learning Outcomes**

Students shall be able to;

1. Model complex systems using mathematical and quantitative methods.
2. Identify and explain commonly used OR techniques including modeling and decision analysis
3. Create their own formulations and expand existing formulations
4. Critically evaluate the impact of model assumptions
5. Choose an appropriate solution technique for a given formulation.

**Course Content**

Introduction to Operation Research: overview of OR, its applications, and its history.Linear Programming: formulation and solution of linear programming problems, graphical solution methods, simplex method, duality theory, sensitivity analysis, and transportation problems. Network Analysis: critical path method (CPM) and the program evaluation and review technique (PERT), which are used to analyze complex projects and identify the critical path. Integer Programming: formulation and solution of integer programming problems, branch and bound algorithm, and cutting plane methods. Nonlinear Programming: gradient-based methods, Newton's method, and constrained optimization. Game Theory: zero-sum games, non-zero-sum games, and cooperative games. Decision Analysis. Network error

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.

**Bayero University Kano (BUK)**

**Computing**

**Department of Software Engineering**

**B.Sc. Software Engineering**

**BUK-SEN 409 Game Design and Development (2 Unit; Core; LH=15, PH=45)**

**Senate-approved relevance**

This course is designed in line with the vision and mission of Bayero University Kano to provide high quality instruction to our students, and equip them with the state-of-the-art knowledge and skills in computer science that they need to take up real-world challenges. To conduct cutting-edge research in areas of national need, frequently in collaboration with other disciplines.

**Overview**

This course provides an introduction to game design and development, covering the essential concepts, tools, and techniques involved in creating successful video games. Students will learn the principles of game design, game mechanics, game engines, scripting languages, graphics, animation, sound design, and user interface design.

The course covers game development history, platforms, goals and genres, player elements, story and character development, gameplay, levels, interface, audio, development team roles, game development process, and marketing and maintenance. Students will play games, analyze them, and complete portions of game designs with appropriate documentation

**Objectives**

The objectives of the course are to:

1. Discuss game mechanics and how they create player engagement
2. Develop proficiency in game engines and programming languages commonly used in game development
3. Explore different game genres and styles, and understand how to create games that appeal to specific audiences
4. Prepare student on how to work collaboratively in a team environment to design, prototype, and playtest a game
5. Discuss the business side of game development, including marketing, monetization, and intellectual property law

**Learning Outcomes**

Students shall be able to;

1. Analyze and identify effective game mechanics, and use them to create engaging gameplay experiences.
2. Design and implement game systems using industry-standard software and programming languages.
3. Create games that fit within specific genres or styles, and understand how to tailor gameplay mechanics and aesthetics to appeal to different audiences.
4. Effectively communicate and collaborate with team members, and use agile development processes to create and refine a game prototype.
5. Develop a basic understanding of the business considerations involved in game development, and be able to create a marketing plan and monetization strategy for a game project.

**Course Content**

Introduction to Game Design and Development**;** what is game design? History of video games**,** Overview of game development process**,** Roles in game development**.** Game Design Fundamentals**;** Game mechanics and rules**,** Game balance and difficulty**,** Game flow and pacing**,** Game narratives and storytelling**,** Game genre and target audience**.** Game Design Tools and Techniques**;** Paper prototyping and playtesting**,** Digital prototyping and game engines**,** Level design and game assets creation**,** User experience and interface design**,** Sound and music in games**.** Game Programming Basics**,** Programming languages and frameworks**,** Object-oriented programming**,** Game physics and collision detection**,** Artificial intelligence and game logic**.** Game Development Process**;** Project management and planning**,** Agile and Scrum methodologies**,** Iterative development and testing**,** Debugging and optimization**.** Advanced Game Design and Development Topics**;** Multiplayer game design and networking**,** Mobile game development, Virtual reality and augmented reality games**,** Game monetization and marketing**,** Game analytics and data-driven design**.** Final Project and Presentation

**Minimum Academic Standard**

Software Engineering programme’s NUC-MAS requirement facilities.