**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 1600 Introduction to Computing Concepts**

**Semester and Level: Semester 1 Level 1**

**Pre-requisites: None**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course presents an overview of computing technology and the field of computer science. Discussion topics will include the organization of modern computers, operating systems, algorithms, programming languages and database systems.

**2. Rationale**

This course provides a fundamental understanding of the Computer Science discipline by focusing on the computers role in representing, storing, manipulating, organizing and communicating data and information. It will provide students with a foundation for future courses in Computer Science and Information Technology. In order to understand the capabilities of computer systems, students must recognize how the various forms of data are stored digitally, how the major hardware components store and operate on such data, and how software is developed and used to control the various subsystems. Students will therefore be introduced to many areas of Computer Science such as computer architecture, operating systems, algorithms, programming languages and database systems.

**3. Course Aims**

This course aims to provide a general introduction of essential computing concepts, techniques and foundations. The course teaches students how to think algorithmically and solve problems efficiently. Students are exposed to core topics, programming languages and system covered by the courses in Computer science and Information Technology in the undergraduate degree programmes.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Analyse a simple circuit involving AND, OR and NOT gates to determine what input bit pattern will produce a specified output.

2. Convert numerical data from one format to various other format

3. Describe the storage hierarchy of data across memory and devices

4. Determine the internal representation of an integer, a character or a floating-point number

5. Describe the execution of an instruction in a classical computer architecture

6. Trace a given simple machine language program to determine the task performed

7. Differentiate between the various ways a processor can communicate with I/O devices

8. Explain operating system concepts such as scheduling, swapping, paging and virtual memory

9. Trace through simple iterative and recursive algorithms

10. Distinguish between procedural, object-oriented and declarative programming paradigms

11. Perform relational operations such as select, join and project on database tables

12. Demonstrate the role of digital logic, control flow and sequencing in computer design

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1.Algorithmic Strategies  
1.1 Divide-and-conquer (cross-reference SDF/Algorithms and Design/Problem-solving strategies)  
2.Assembly Level Machine Organization  
2.1 Basic organization of the von Neumann machine  
2.2 Control unit; instruction fetch, decode, and execution  
2.3 Instruction sets and types (data manipulation, control, I/O)  
2.4 Assembly/machine language programming  
2.5 Instruction formats  
2.6 Addressing modes  
2.7 Subroutine call and return mechanisms (cross-referencePL/Language Translation and Execution)  
2.8 I/O and interrupts  
2.9 Shared memory multiprocessors/multicore organization  
2.10 Introduction to SIMD vs. MIMD and the Flynn Taxonomy  
3.Basic Analysis  
3.1 Analysis of iterative and recursive algorithms  
4.Computational Paradigms  
4.1 Basic building blocks and components of a computer (gates, flip-flops, registers, interconnections;  
5.Concurrency  
5.1 States and state diagrams (cross-reference SF/StateandState Machines)  
5.2 Structures (ready list, process control blocks, and so forth)  
5.3 Dispatching and context switching  
6.Device Management  
6.1 Characteristics of serial and parallel devices  
6.2 Direct memory access  
7.DigitalLogic andDigitalSystems  
7.1 Overview and history of computer architecture  
7.2 Multiple representations/layers of interpretation (hardware is just another layer)  
8.File Systems  
8.1 Files: data, metadata, operations, organization, buffering, sequential, nonsequential  
9.Functional Organization  
9.1 Instruction pipelining  
10.Fundamental Data Structures and Algorithms  
10.1 Sequential and binary search algorithms  
10.1 Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max,  
11.History  
11.1 History of computer hardware, software, networking (cross-reference AR/Digital logic and digital systems/  
12.Interfacing andCommunication  
12.1 I/O fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O  
12.2 External storage, physical organization, and drives  
12.3 Buses: bus protocols, arbitration, direct-memory access (DMA)  
13.Language Translation and Execution  
13.1 Interpretation vs. compilation to native code vs. compilation to portable intermediate representation  
13.2 Memory management  
13.3 Programming paradigms: Procedural, Object-oriented and declarative programming  
14.Machine Level Representation of Data  
14.1 Bits, bytes, and words  
14.2 Numeric data representation and number bases  
14.3 Fixed-and floating-point systems  
14.4 Signed and twos-complement representations  
14.5 Representation of non-numeric data (character codes, graphical data)  
14.6 Representation of records and arrays  
14.7 Data representation: characters, integers, floating-point numbers  
15.Memory Management  
15.1 Review of physical memory and memory management hardware  
15.2 Working sets and thrashing  
16.Memory System Organization and Architecture  
16.1 Storage systems and their technology  
16.2 Memory hierarchy: importance of temporal and spatial locality  
16.3 Main memory organization and operations  
16.4 Latency, cycle time, bandwidth, and interleaving  
16.5 Cache memories (address mapping, block size, replacement and store policy)  
16.6 Virtual memory (page table, TLB)  
17.Multiprocessing and Alternative Architectures  
17.1 Example SIMD and MIMD instruction sets and architectures  
18.Operating System Principles  
18.1 Structuring methods (monolithic, layered, modular, micro-kernel models)  
18.2 Device organization  
18.3 Concept of user/system state and protection, transition to kernel mode  
19.Overview of Operating Systems  
19.1 Role and purpose of the operating system  
19.2 Functionality of a typical operating system  
20.Processing  
20.1 Fundamental programming concepts:  
21.Query Languages  
21.1 Select-project-join  
22.Security and Protection  
22.1 Overview of system security  
22.2 Policy/mechanism separation  
23.State and State Machines  
23.1 Simple logic gates, logical expressions, Boolean logic simplification  
24.Virtual Machines  
24.1 Paging and virtual memory

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Lectures with pauses for students to answer MCQ based on preceeding content. |
| Online Activities (myElearning) | Unit Review Quizzes - formative |
| Laboratory Work | Problem Solution: Algorithms, Flowcharts and Deskchecking (Tracing) |
| Flipped Classroom | Case Studies to solve and present solution to the class |
| Inquiry-based Learning | Structured problems to strengthen concept understanding |
| Problem-based Learning | Groups of students placed in breakout rooms for set time period to work on cases/problems; present solution to class; discussion should follow. |
| Groupwork | Conducted during live lectures; Use of breakout rooms |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

In this course, assessments are designed to test student knowledge and problem solving skills through a combination of theoretical/analytical exercises and practical software-related problem-solving tasks. There are five coursework assessments in this course worth 100% collectively: 4 individual assignments based on theoretical and practical concepts and 1 group project.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Assignment 1 Number Systems and Logic circuit problems to solve. | X | X | X | X |  |  |  |  |  |  |  |  | 15% | Problem Solving: structured questions | 10-14 days |
| Assignment 2 Machine Language Program trace; Operating Systems Problems |  |  |  |  | X | X | X | X |  |  |  |  | 20% | Problem Solving: structured questions | 10-14 days |
| Assignment 3 Algorithms: Sequence & Selection |  |  |  |  |  |  |  |  |  |  |  |  | 20% | Problem Solving: Algorithms, Flowcharts and Deskchecking (Tracing) | 10-14 days |
| Assignment 4 Algorithms: Iteration & Recursion; Programming Languages |  |  |  |  |  |  |  |  | X | X |  |  | 20% | Problem Solving: Algorithms, Flowcharts and Deskchecking (Tracing) | 10-14 days |
| Project: Database - SQL; Case study: Computer System Design |  |  |  |  |  |  |  |  |  |  | X | X | 25% | Case Studies to review, scenario analysis, solution design and implementation | 14 days |
| Total % |  | | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Computer Science, an Overview. (2018) Glenn Brookshear & Dennis Brylow, 13th edition. Pearson Publishing. ISBN: 978-0134875460\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,Lecture notes, handouts, content provided on myElearning LMS,

,,

NONE,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Data Storage: Data representation. | Chapter 1 - Data Structure | 1, 2 |  |  |
| 2. | Data Storage: Primary and secondary storage |  | 5, 6 | Assignment 1 | Week 3 |
| 3. | Data manipulation: Parts of CPU. Machine language and machine cycle. | Chapter 2. Data Manipulation | 1, 2 |  |  |
| 4. | Data manipulation: Program execution. Arithmetic and logic instructions. |  | 5, 6 | Assignment 2 | Week 5 |
| 5. | Main memory and communication of CPU with other devices. Other machine architectures. | Chapter 3. Operating Systems | 1, 2 |  |  |
| 6. | Operating Systems: Introduction to the theory of operating systems and basic concepts. |  | 6, 7 | Assignment 3 | Week 7 |
| 7. | Algorithms: Basic concepts. Algorithm representation. | Chapter 5 Algorithms | 1, 2 |  |  |
| 8. | Algorithms: Practice Exercises | Handout | 3, 5, 6 | Assignment 4 | Week 10 |
| 9. | Database Basic concepts. Relational model and SQL. | Chapter 9  Databases | 1, 2 |  |  |
| 10. | Database: Practice Exercises | Handout | 3, 5, 6 |  |  |
| 11. | Programming Languages | Chapter  6  - Programming Languages | 1, 2, 4 | Project | Week 12 |
| 12. | Programming Languages | Handout | 5, 6 |  |  |
| 13. | Course Review(No introduction of new subject matter) |  | 5, 6 |  |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 1601 Computer Programming I**

**Semester and Level: Semester 1 Level 1**

**Pre-requisites: None**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course uses an appropriate, contemporary programming language as a tool to teach fundamental programming concepts, topics and techniques. The main concepts covered are sequence, selection and repetition logic, character and string manipulation, functions, and a basic introduction to arrays and their applications.

**2. Rationale**

This course equips first year undergraduate students with the skills to solve problems on computer based systems. It identifies what type of problems can be solved by such systems and which cannot. The focus in this course is problem description, definition and presentation using either flowcharting or pseudocode tools. By guiding students towards developing structured algorithms using well known methods and techniques, the course lays the foundation for building competency in programming.

**3. Course Aims**

This course aims to provide an introduction to programming. The course teaches students how to implement algorithmic solutions for common problems and tasks. Students are exposed to a contemporary programming language and learn how to write, test and debug programs.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Discuss the importance of algorithms in the problem-solving process.

2. Discuss how a problem may be solved by multiple algorithms, each with different properties.

3. Create algorithms for solving simple problems.

4. Use a programming language to implement, test, and debug algorithms for solving simple problems.

5. Analyse and explain the behaviour of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion.

6. Identify, describe and write programs that use primitive data types.

7. Modify and expand short programs that use standard conditional and iterative control structures and functions.

8. Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing.

9. Write a program that uses file I/O to provide persistence across multiple executions.

10. Choose appropriate conditional and iteration constructs for a given programming task.

11. Describe the concept of recursion and give examples of its use.

12. Identify the base case and the general case of a recursively-defined problem.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. The concept and properties of algorithms  
2. Informal comparison of algorithm efficiency (e.g., operation counts)  
3. The role of algorithms in the problem-solving process  
4. Basic syntax and semantics of a higher-level language   
5. Variables and primitive data types (e.g., numbers, characters, Booleans, strings)  
6. Expressions and assignments  
7. Conditional and iterative control structures  
8. Simple I/O including file I/O  
9. Functions and parameter passing  
10. One dimensional arrays  
11. The concept of recursion

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered twice weekly |
| Online Activities (myElearning) | Quizzes |
| Laboratory Work |  |
| Flipped Classroom | Readings from text book.Video lectueres by the author of the essential textbook |
| Inquiry-based Learning | Interactive tutorial sessions - students solve problems on the board |
| Problem-based Learning | Apply programming concepts to real world problems |
| Groupwork |  |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

3 Assingments, 3 online quizzes, 2 courswork exams, 1 final exam

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Assignment 1 | X | X | X |  |  |  |  |  |  |  |  |  | 5 | Take home programming problems | 10-14 days |
| Assignment 2 |  |  | X | X | X |  | X | X |  |  | X |  | 6 | Take home programming problems | 10-14 days |
| Assignment 3 |  |  |  |  |  |  |  |  |  |  |  |  | 6 | Take home programming problems | 10-14 days |
| Quiz 1 | X | X | X | X | X |  |  |  |  |  |  |  | 1 | Online quiz, MCQ | 15-30 min |
| Quiz 2 |  |  | X | X | X | X | X |  |  |  | X |  | 1 | Online quiz, MCQ | 15-30 min |
| Quiz 3 |  |  |  |  | X | X |  |  | X |  | X |  | 1 | Online quiz, MCQ | 15-30 min |
| Coursework Exam 1 |  |  | X | X | X |  | X | X |  |  | X |  | 10 | Problems & short answer questions | 1.5 hours |
| Coursework Exam 2 |  |  | X | X | X | X | X | X | X | X | X | X | 20 | Problems & short answer questions | 1.5 hours |
| Final Examinaiton | X | X | X | X | X | X | X | X | X | X | X | X | 50 | Problems & short answer questions | 2 hours |
| Total % |  | | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Noel Kalicharan. C Programming - A Beginner's Course. 2005. CreateSpace Independent Publishing Platform.,,

,,

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | 1, 2 | C Programming - A Beginner's Course: Chapter 1 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 2. | 2, 3 | C Programming - A Beginner's Course: Chapter 1 | Lectures, Tutorials, Worksheets, online videos | Assignment 1 | WK2 |
| 3. | 4 | C Programming - A Beginner's Course: Chapter 2 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 4. | 5 | C Programming - A Beginner's Course: Chapter 2 and 6 | Lectures, Tutorials, Worksheets, online videos | Quiz 1 | WK4 |
| 5. | 6 | C Programming - A Beginner's Course: Chapter 2 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 6. | 7 | C Programming - A Beginner's Course: Chapter 3 | Lectures, Tutorials, Worksheets, online videos | Assignment 2 | WK6 |
| 7. | 8 | C Programming - A Beginner's Course: Chapter 6 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 8. | 9 | C Programming - A Beginner's Course: Chapter 7 | Lectures, Tutorials, Worksheets, online videos | Coursework Exam 1 | WK8 |
| 9. | 9 | C Programming - A Beginner's Course: Chapter 7 | Lectures, Tutorials, Worksheets, online videos | Quiz 2 | WK9 |
| 10. | 10 | C Programming - A Beginner's Course: Chapter 8 | Lectures, Tutorials, Worksheets, online videos | Assignment 3 | WK10 |
| 11. | 10, 11 | C Programming - A Beginner's Course: Chapter 8 | Lectures, Tutorials, Worksheets, online videos | Quiz 3 | WK11 |
| 12. | 11 | C Programming - A Beginner's Course: Chapter 8 | Lectures, Tutorials, Worksheets, online videos | Coursework Exam 2 | WK12 |
| 13. | Course Review(No introduction of new subject matter) |  |  |  |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 1602 Computer Programming II**

**Semester and Level: Semester 2 Level 1**

**Pre-requisites: None**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course uses an appropriate programming language as a tool to teach intermediate programming concepts. The main concepts covered are structures, one and two dimensional arrays and applications involving searching, sorting and merging, random number generation, numerical methods, games and simulation.

**2. Rationale**

This course introduces new programming concepts. It focuses on the use of common searching and sorting methods and character and string manipulation which are two very important concepts that are necessary in most software systems.

**3. Course Aims**

This course aims to introduce students to intermediate programming concepts.This course teaches students the foundations of modeling, searching, sorting, and handling multi-dimensional data. Mastery of these essential topics in Computer Science allows students to extend the capabilities of a given programming language to meet the needs of a specific problem.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Implement basic numerical algorithms.

2. Implement simple search algorithms and explain the differences in their time complexities.

3. Apply Bayes rule to determine the probability of a hypothesis given evidence.

4. Explain how conditional independence assertions allow for greater efficiency of probabilistic systems.

5. Use random number generation in simulation and game development.

6. Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion.

7. Discuss the appropriate use of built-in data structures.

8. Describe common applications for each of the following data structures: stack, queue, priority queue, set, and map.

9. Design programs which utilize one and two-dimensional arrays.

10. Write programs that use each of the following data structures: arrays, records/structs, and strings.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Arrays   
1.1. Declaring one-dimensional arrays   
1.2. Finding largest and smallest values   
1.3. Functions and parameter passing  
1.4. Passing arrays as arguments   
2. Structures   
2.1. Declaration   
2.2. User Defined Types   
2.3. Array of Structures   
2.4. Nested Structures   
3. Strings  
3.1 C-Strings and Strings  
3.2. Array of Strings   
3.3. Strings and string processing  
4. Sorting, Searching and Merging   
4.1. Sorting arrays   
4.2. Inserting elements in place   
4.3. Sorting parallel arrays   
4.4. Insertion Sort, Selection Sort, Bubble Sort   
4.5. Linear Search  
4.6. Binary Search  
4.7. Merging ordered lists, Merge Sort  
5. Two dimensional arrays   
5.1. 2 dimensional arrays: concept, creating, adding data   
5.2. 2 dimensional arrays: traversing column only, row only, specific areas of the 2D array   
5.3. 2 dimensional arrays: Manipulating   
6. Random Numbers   
6.1. Random and pseudo random numbers   
6.2. Random number generation and ranges   
6.3. Axioms of probability   
6.4. Probabilistic inference   
6.5. Bayes Rule  
7. Games and Simulation   
7.1. Simulation of real-life problems   
7.2. Simulating a queue   
8. Files  
8.1. Reading data from a file  
8.2. Sending output to a file  
8.3. Binary files  
8.4. Text file vs. Binary file  
8.5. Opening and closing files  
8.6. Random Access files  
8.7. Indexed files

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Online Activities (myElearning) | Quizzes |
| Laboratory Work |  |
| Flipped Classroom | Readings from text book.Video lectueres by the author of the essential textbook |
| Inquiry-based Learning | Interactive tutorial sessions - students solve problems on the board |
| Problem-based Learning | Apply programming concepts to real world problems |
| Groupwork |  |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |
|  |  |  |  |

**8. Course Assessments Description**

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Assignment 2 |  |  |  |  | X |  | X | X |  | X | 6 | Take home programming problems | 10-14 days |
| Assignment 3 |  |  | X |  | X |  |  |  | X |  | 6 | Take home programming problems | 10-14 days |
| Quiz 1 |  |  |  |  |  |  |  |  |  |  | 1 | Online quiz, MCQ | 15-30 min |
| Quiz 2 |  |  | X | X | X |  |  |  | X |  | 1 | Online quiz, MCQ | 15-30 min |
| Quiz 3 | X | X |  |  |  |  |  |  | X |  | 1 | Online quiz, MCQ | 15-30 min |
| Coursework Exam 1 | X |  |  |  | X | X | X | X |  | X | 10 | Problems & short answer questions | 1.5 hours |
| Coursework Exam 2 | X | X | X | X | X | X | X | X | X | X | 20 | Problems & short answer questions | 1.5 hours |
| Final Examination | X | X | X | X | X | X | X | X | X | X | 50 | Problems & short answer questions | 2 hours |
| Total % |  | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

,,Recommended

Brian W. Kernighan, Dennis M. Ritchie. C the programming language 2nd edition. 1988. Pearson.,Bjarne Stroustrup. The C ++ Programming language. 2013. Addison-Wesley Professional.,Other: (Special Equipment/Tools)

,11. Staffing Requirements

**11. Staffing Requirements:**

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 2. | Arrays 1.3, 1.4 | C Programming - A Beginner's Course, Chapter 8 | Lectures, Tutorials, Worksheets, online videos | Assignment 1 | Week 2 |
| 3. | Structures 2.1, 2.2, 2.4 | C Programming - An Advanced Course, Chapter 2 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 4. | Structures 2.3, Files 8.1 - 8.5 | C Programming - An Advanced Course, Chapter 2 | Lectures, Tutorials, Worksheets, online videos | Quiz 1 | Week 4 |
| 5. | Files 8.6, 8.7, Strings 3.1 | C Programming - An Advanced Course, Chapter 8, C Programming - A Beginner's Course, Chapter 2 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 6. | Strings 3.2, 3.3 | C Programming - A Beginner's Course, Chapter 2 | Lectures, Tutorials, Worksheets, online videos | Coursework exam 1 | Week 6 |
| 7. | Random Numbers 6.1 - 6.4 | C Programming - An Advanced Course, Chapter 7 | Lectures, Tutorials, Worksheets, online videos | Assignment 2 | Week 7 |
| 8. | Random Numbers 6.5, 7.1, 7.2 | C Programming - An Advanced Course, Chapter 7 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 9. | Two dimensional arrays 5.1, 5.2, 5.3 | C Programming - An Advanced Course, Chapter 1 | Lectures, Tutorials, Worksheets, online videos | Assignment 3, Quiz 2 | Week 9 |
| 10. | Sorting, Searching and Merging 4.1, 4.2, 4.3 | C Programming - An Advanced Course, Chapter 1 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 11. | Sorting, Searching and Merging 4.4, 4.5 | C Programming - An Advanced Course, Chapter 1 | Lectures, Tutorials, Worksheets, online videos | Quiz 3 | Week 11 |
| 12. | Sorting, Searching and Merging 4.6, 4.7 | C Programming - An Advanced Course, Chapter 1 | Lectures, Tutorials, Worksheets, online videos | Coursework Exam 2 | Week 12 |
| 13. | Course Review(No introduction of new subject matter) |  |  | X |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 1603 Computer Programming III**

**Semester and Level: Semester 2 Level 1**

**Pre-requisites: None**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course uses an appropriate programming language as a tool to teach intermediate programming concepts. The main concepts covered are pointers, linked lists, stacks, queues, sets. and maps and their implementations using arrays and linked lists and recursion.

**2. Rationale**

This course introduces intermediate programming concepts and abstract data types. It focuses on the use of common abstract data types such as linked lists, stacks, queues, sets, and maps which are important concepts that are inherent in most systems. It introduces the concept of dynamic storage methods which are necessary in programming easily adaptive systems. It also covers the conditions under which dynamic storage is applicable and when it is not.

**3. Course Aims**

This course aims to introduce students to pointers and abstract data types.This course teaches students the foundations of working with dynamic allocations, interfaces, and re-use of code. Mastery of these essential topics in Computer Science allows students to design and implement memory-efficient data types and solutions when meeting the needs of a specific problem.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Choose the appropriate data structure for modeling a given problem.

2. Design and use the following data structures: arrays, records/structs, strings, linked lists, stacks, queues, sets, and maps to model real world objects and concepts.

3. Write programs that use each of the following data structures: arrays, records/structs, strings, linked lists, stacks, queues, sets, and maps.

4. Compare alternative implementations of data structures with respect to performance.

5. Explain what a pointer is and how it can be used.

6. Describe how references allow for objects to be accessed in multiple ways.

7. Compare and contrast the costs and benefits of dynamic and static data structure implementations.

8. Identify common coding errors that lead to insecure programs (e.g., buffer overflows, memory leaks, malicious code) and apply strategies for avoiding such errors.

9. Conduct a personal code review (focused on common coding errors) on a program component using a provided checklist.

10. Use recursion as a powerful technique in problem solving.

11. Identify the base case and the general case of a recursively-defined problem.

12. Apply a variety of strategies to the testing and debugging of simple programs.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Records/structs (heterogeneous aggregates)  
1.1. Declaration   
1.2. User Defined Types   
1.3. Array of Structures   
1.4. Nested Structures   
2. Pointers   
2.1. Concept of a pointer   
2.2. Pointers as arguments   
2.3. Pointer arithmetic   
3. Linked Lists   
3.1. Operations on a Linked List (inserting, deleting, counting, searching)   
3.2. Allocating Memory (e.g., sizeof, malloc, calloc, and free in C)   
3.3. Sorted Lists   
3.4. Merging   
4. Abstract data types and their implementation  
4.1. Stacks (Array and Linked List Implementation)   
4.1.1. Creating a Stack Header File   
4.1.2. How to convert from infix to postfix   
4.2. Queues (Array and Linked List Implementation)   
4.3. Priority queues (Array and Linked List Implementation)   
4.4. Sets  
4.5. Maps  
5. Recursion  
5.1. The concept of recursion   
5.2. Recursive functions   
5.3. Recursive applications

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered twice weekly |
| Online Activities (myElearning) | Quizzes |
| Laboratory Work |  |
| Flipped Classroom | Readings from text book.Video lectueres by the author of the essential textbook |
| Inquiry-based Learning | Interactive tutorial sessions - students solve problems on the board |
| Problem-based Learning | Apply programming concepts to write space-efficient programs for real world problems |
| Groupwork |  |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

3 Assingments, 3 online quizzes, 2 courswork exams, 1 final exam

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Assignment 1 | X | X |  |  |  |  | X |  | X |  |  | X | 5 | Take home programming problems | 10-14 days |
| Assignment 2 | X | X | X |  | X | X | X |  | X |  |  | X | 6 | Take home programming problems | 10-14 days |
| Assignment 3 |  |  |  |  |  |  |  |  |  |  |  |  | 6 | Take home programming problems | 10-14 days |
| Quiz 1 | X | X |  |  | X |  |  |  |  |  |  |  | 1 | Online quiz, MCQ | 15-30 min |
| Quiz 2 | X | X | X | X | X | X | X | X |  |  |  |  | 1 | Online quiz, MCQ | 15-30 min |
| Quiz 3 | X | X | X | X | X | X | X | X |  | X |  |  | 1 | Online quiz, MCQ | 15-30 min |
| Coursework Exam 1 | X | X | X | X | X | X | X |  |  |  |  |  | 10 | Problems & short answer questions | 1.5 hours |
| Coursework Exam 2 | X | X | X | X | X | X | X | X | X | X | X |  | 20 | Problems & short answer questions | 1.5 hours |
| Final Examination | X | X | X | X | X | X | X | X | X | X | X |  | 50 | Problems & short answer questions | 2 hours |
| Total % |  | | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Noel Kalicharan. C Programming - An Advanced Course. 2006. CreateSpace Independent Publishing Platform.,Robert Sedgewick. Algorithms. 1983. Addison-Wesley.,

Noel Kalicharan. Advance Topics in C: Croe Concepts in Data Structures (Expert's Voice in C). 2013. Apress.,,

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Structures 1.1 - 1.4 | C Programming - An Advanced Course, Chapter 2 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 2. | Pointers 2.1, 2.2 | C Programming - An Advanced Course, Chapter 3 | Lectures, Tutorials, Worksheets, online videos | Assignment 1 | Week 2 |
| 3. | Pointers 2.2, 2.3 | C Programming - An Advanced Course, Chapter 3 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 4. | Linked List 3.1, 3.2 | C Programming - An Advanced Course, Chapter 4 | Lectures, Tutorials, Worksheets, online videos | Quiz 1 | Week 4 |
| 5. | Linked List 3.3, 3.4 | C Programming - An Advanced Course, Chapter 4 | Lectures, Tutorials, Worksheets, online videos | Assignment 2 | Week 5 |
| 6. | Stacks 4.1 | C Programming - An Advanced Course, Chapter 5 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 7. | Stacks 4.1, Queues 4.2 | C Programming - An Advanced Course, Chapter 5, Robert Sedgewick. Algorithms, Chapter 11 | Lectures, Tutorials, Worksheets, online videos | X |  |
| 8. | Queues 4.2, Priority queues 4.3 | Robert Sedgewick. Algorithms, Chapter 11 | Lectures, Tutorials, Worksheets, online videos | Coursework Exam 1 | Week 8 |
| 9. | Priority queues 4.3, Sets 4.4 | Robert Sedgewick. Algorithms, Chapter 11, Chapter 30 pgs 308 - 405 | Lectures, Tutorials, Worksheets, online videos | Quiz 2 | Week 9 |
| 10. | Sets 4.4, Maps 4.5 | Robert Sedgewick. Algorithms, Chapter 30 pgs 308 - 405 | Lectures, Tutorials, Worksheets, online videos | Assignment 3 | Week 10 |
| 11. | Recursion 5.1, 5.2 | C Programming - An Advanced Course, Chapter 6 | Lectures, Tutorials, Worksheets, online videos | Quiz 3 | Week 11 |
| 12. | Recursion 5.3 | C Programming - An Advanced Course, Chapter 6 | Lectures, Tutorials, Worksheets, online videos | Coursework Exam 2 | Week 12 |
| 13. | Course Review(No introduction of new subject matter) |  |  | X |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 1604 Mathematics for Computing**

**Semester and Level: Semester 2 Level 1**

**Pre-requisites: None**

**Co-requisites: None**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course introduces students to the basic mathematical structures and computational techniques that are considered to be the foundation for courses in Computer Science and Information Technology. Students are also taught how to reason logically and how to solve problems using various proof techniques. The main mathematical structures covered are logic, sets, relations, functions and counting techniques.

**2. Rationale**

The material in this course is pervasive in many areas in computing such as data structures, algorithms, security, databases and programming languages. The ability to reason logically is important in order to write correct computer programs and students are also expected to create and understand proofs in courses such as data structures, algorithms, intelligent systems and information security.

**3. Course Aims**

This course has the following primary learning goals. These goals are developed in almost all the course topics. Familiarity with the role logical reasoning plays in mathematics: Students should become aware of the basic principles of logical reasoning. They should come to understand the role of precise definitions for deciding whether a given object satisfies a given mathematical term, appreciate the importance of knowing both examples and non-examples of mathematical terms in order to explore statements about them, and know how to use definitions effectively in proofs. They should be familiar with the basic structures of direct proofs, indirect proofs, and proofs by mathematical induction and be able to apply them in a variety of situations. They should be able both to follow elementary mathematical arguments and to identify mistakes in them. Experience with mathematical exploration and conjecture: Students should experience the challenge and pleasure of exploring new concepts, looking for patterns, making conjectures, and trying to decide whether they are true or false. Communication: Students should be able to discuss mathematical ideas coherently and express themselves clearly when giving a proof or counterexample. Connections with applications: Students should come to appreciate connections between mathematical concepts (such as those between general relations on sets and functions or between propositional logic and set properties) and between abstract mathematical ideas and concrete applications.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Simplify and evaluate basic logic statements including compound statements, implications,inverses, converses, contrapositives and quantifiers using truth tables and the properties of logic

2. Apply the concepts of standard mathematical logic to produce proofs or refutations of well-formed propositions or arguments, and to determine the validity of arguments

3. Demonstrate the ability to write and evaluate a proof or outline the basic structure of and giveexamples of each proof technique described.

4. Explain the relationship between weak and strong induction and give examples of the appropriate use of each.

5. Use and analyse recursive relations

6. Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures.

7. Formulate problems in the language of sets, functions & relations and perform the associatedoperations

8. Use elementary number theory including the divisibility properties of numbers to determineprime numbers and composites, the greatest common divisor, and the least common multiple;perform modulo arithmetic and computer arithmetic.

9. Solve counting problems by applying elementary counting techniques using the product andsum rules, permutations, combinations, the pigeon-hole principle, and binomial expansion.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Propositional logic  
1.1. Logical connectives  
1.2. Truth tables  
1.3. Normal forms (conjunctive and disjunctive)  
1.4. Validity of well-formed formula  
1.5. Propositional inference rules (concepts of modus ponens and modus tollens)   
1.6. Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction  
1.7. Limitations of propositional logic   
2. Predicate logic  
2.1. Universal and existential quantification  
2.2. Limitations of predicate logic   
3. Basic Proof Techniques   
3.1.The structure of mathematical proofs  
3.2.Direct proofs  
3.3.Disproving by counterexample   
3.4.Proofs by contradiction and contraposition (i.e. Indirect Proofs)  
3.5.Proofs of Elementary Number Theory Results  
3.6. Results on Rational and Irrational Numbers   
4. Principles of Mathematical Induction and Recursive Definitions  
4.1. Weak and strong induction (i.e., First and Second Principle of Induction)  
4.2. Recursive mathematical definitions  
4.3. Sequences  
4.4. Structural induction   
5. Sets: Union,Intersection, complement, Cartesian product  
5.1. Power sets  
5.2. Cardinality of finite sets   
6. Relations and Modular Arithmetic  
6.1. Graphs  
6.2. Reflexivity, Symmetry, Transitivity, Equivalence relations   
6.3. Basic Modular Arithmetic  
6.4. Partial orders   
7. Functions: Surjections, Injections, Bijections, Inverses, Composition   
8. Counting  
8.1. Set cardinality and Counting  
8.2. Sum and Product rules  
8.3. Inclusion-Exclusion principle  
8.4. Pigeonhole principle  
8.5. Permutations and combinations  
8.6. Pascals identity  
8.7. Binomial theorem  
8.8. Recurrence Relations

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered twice weekly |
| Interactive Tutorials | Live tutorial held once per weekly (for each tutorial group) |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Tutorial | 13 | 1 hours | 1 hours |
| Total: | 13 | 3 hours | 3 hours |

**8. Course Assessments Description**

Six assessments including 3 assignments, 2 Incourse Exams, and 1 Final Exam In Course Assessment has a weight of 50 %, Final Exam has a weight of 50%

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Assignment 1 | X | X | X |  |  |  |  |  |  | 3 | Short Answer and Problem Solving Exercises | 10-14 days |
| Assignment 2 | X | X | X | X | X |  |  |  |  | 3 | Short Answer and Problem Solving Exercises | 10-14 days |
| Assignment 3 |  |  |  |  |  |  |  |  |  | 4 | Short Answer and Problem Solving Exercises | 10-14 days |
| Coursework Exam 1 | X | X | X | X | X |  |  |  |  | 20 | MCQ, Short Answer Questions and Problem Solving Exercises | 60-90 minutes |
| Coursework Exam 2 | X | X | X | X | X | X | X |  |  | 20 | MCQ, Short Answer Questions and Problem Solving Exercises | 60-90 minutes |
| Final Exam | X | X | X | X | X | X | X | X | X | 50 | Short Answer and Problem Solving Exercises | 2 hours |
| Total % |  | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

\* Epp, S - Discrete Mathematics with Applications, 5th edition. 2019. Cengage Learning.,,

Doer, Al and Ken Levasseur - Applied Discrete Structures (freely available online at https://discretemath.org/ads/index-ads.html),Rosen, Kenneth - Discrete Mathematics and Its Applications 8th Edition. 2018. McGraw Hill.,Grimaldi, Ralph P. - Discrete and Combinatorial Mathematics: An Applied Introduction, 5th Edition. 2003. Pearson.

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Propositional Logic | Sec 2.1 - 2.2, Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | X | X |
| 2. | Propositional Logic | Sec 2.2 - 2.4, Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Assignment 1 Given | X |
| 3. | Predicate Logic | Sec 3.1 - 3.2, Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Assignment 1 Due | Week 2 - Week 3 |
| 4. | Predicate logic and Basic Proof Techniques | Sec 3.4, 4.1, 4.2 Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | X | X |
| 5. | Basic Proof Techniques | Sec 4.3, 4.6, 4.7 Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Assignment 2 Given | X |
| 6. | Principles of Mathematical Induction | Sec 5.1 - 5.3, Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Assignment 2 Due | Week 5 - Week 6 |
| 7. | Recursive Definitions, Sets | Sec 5.6, 5.7, 6.1, Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | X | X |
| 8. | Sets, Relations | Sec 6.2, 6.3, 8.1 Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Coursework Exam 1 | Week 8 |
| 9. | Relations and Modular Arithmetic | Sec 8.1 - 8.4, Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Assignment 3 Given | X |
| 10. | Relations, Functions | Sec 7.1 - 7.3 Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Assignment 3 Due | Week 9 - Week 10 |
| 11. | Counting | Sec 9.1 - 9.3 Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | X | X |
| 12. | Counting | Sec 9.4, 9.5, 9.7 Epp, Discrete Math 5e | 2 Lectures and 1 Tutorial | Coursework Exam 2 | Week 12 |
| 13. | Course Review(No introduction of new subject matter) |  | 2 Lectures and 1 Tutorial | X | X |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 2603 Object-Oriented Programming I**

**Semester and Level: Semester 2 Level 2**

**Pre-requisites: COMP 1603**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course provides a comprehensive introduction to the concepts and techniques of object oriented programming. Students will be taught fundamental object oriented constructs such as classes, methods and objects and core concepts such as encapsulation, abstraction, inheritance and composition. Students will learn how to develop user interfaces using an object oriented toolkit (e.g. Swing in Java) and the course gives a preliminary introduction to object oriented design. At the end of the course, students will be able to develop object oriented programs. The course will be delivered using a combination of face-to-face lectures and interactive hands-on computer lab sessions, along with eLearning activities using various online resources. Assessments will take the form of written examinations and individual programming assignments.

**2. Rationale**

An object oriented programming paradigm essentially aims to model real world artefacts in a manner that is modular, reusable and flexible. This course shows students how to break down large modelling problems into sub-problems and develop solutions that feature self contained, modular code.

**3. Course Aims**

COMP2603 aims to develop practical object oriented programming skills in undergraduate students while promoting an understanding of the theoretical concepts and design considerations behind these techniques. The course also aims to expose students to the development tools and programming APIs commonly used in the field.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Describe the fundamental concepts and vocabulary of an object-oriented approach.

2. Analyze a real-world situation in an object oriented way.

3. Understand and interpret designs expressed in UML diagrams .

4. Design object oriented solutions containing multiple classes and collaborations.

5. Implement object oriented models using an appropriate programming language.

6. Develop graphical user interfaces using objects within a programming framework.

7. Generate event-driven code for making graphical user interfaces interactive and functional.

8. Explain the impact of small design changes on object oriented program behaviour/outcomes.

9. Apply basic object oriented techniques to real-world programming problems.

10. Use appropriate collections for data storage and manipulation.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1.Object-Oriented Programming  
1.1.Object-oriented design  
1.1.1.Decomposition into objects carrying state and having behaviour  
1.1.2.Class-hierarchy design for modelling  
1.2.Definition of classes: fields, methods, and constructors  
1.3.Subclasses, inheritance, and method overriding  
1.4.Dynamic dispatch: definition of method-call  
1.5.Subtyping (cross-reference PL/Type Systems)  
1.5.1.Subtype polymorphism; implicit upcasts in typed languages  
1.5.2.Notions of behavioural replacement:subtypes acting like supertypes  
1.5.3.Relationship between subtyping and inheritance  
1.6.Object-oriented idioms for encapsulation and information hiding  
1.6.1.Privacy and visibility of class members  
1.6.2.Interfaces revealing only method signatures  
1.6.3.Abstract base classes  
1.7.Using collection classes, iterators, and other common library components  
1.8.Overloading, overriding of methods, constructors  
1.9.Method types: accessors, mutators, static methods, instance methods  
2.Basic Type Systems  
2.1.Type safety and errors caused by using values inconsistently given their intended types  
2.2.Goals and limitations of static typing  
2.3.Generic types (parametric polymorphism)  
2.4.Complementary benefits of static and dynamic typing  
3.Algorithms and Design  
3.1.Fundamental design concepts and principles  
4.Fundamental Programming Concepts  
4.1.Basic syntax and semantics of a higher-level language  
4.2.Variables and primitive data types (e.g., numbers, characters, Booleans)  
4.3.Expressions and assignments  
4.4.Simple I/O including file I/O  
4.5.Conditional and iterative control structures  
4.6.Functions and parameter passing  
5.Fundamental Data Structures  
5.1.Arrays  
5.2.Strings and string processing  
5.3.Abstract data types and their APIs  
5.4.Strategies for choosing the appropriate abstract data structure  
6.Event-Driven and Reactive Programming  
6.1.Events and event handlers  
6.2.Separation of model, view, and controller  
7.SoftwareDesign  
7.1.System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion, re-use of standard structures  
7.2.Design Paradigms such as structured design (top-down functional decomposition), object-oriented design, event driven design, function oriented, service oriented  
8.Designing Interaction and HCI Foundations  
8.1.Elements of visual design (layout, color, fonts, labeling)  
8.2.Principles of good design and good designers; engineering tradeoffs  
9. Programming Interactive Systems  
9.1.Software Architecture Patterns, e.g.,Model-View controller; command objects, online, offline (cross  
9.2.Event management and user interaction  
9.3.Modern GUI libraries (e.g. JavaFX) GUIbuilders and UI programming environments

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered either online or face-to-face twice weekly |
| Online Activities (myElearning) | Weekly activities: short quiz, design task/programming task/ forum posting task |
| Flipped Classroom | Early release of lecture content, lab exercises with extra activities for debate/discussion/exploration |
| Inquiry-based Learning | Interactive code writing/ demonstrations / problem solving during lab sessions |
| Problem-based Learning | Exercises/activities/tasks from lectures, labs and worksheets |
| Directed Discussions | Online and during lectures and lab sessions |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

"In this course, assessments are designed to test student knowledge and practical skills through a combination of theoretical/analytical exercises and practical software-related problem-solving tasks. There are five coursework assessments in this course worth 50% collectively: 3 programming assignments and 2 written coursework exams based on theoretical and practical concepts. The remaining 50% is assessed in a final 2 hour written examination.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Assignment 1 |  | X | X | X |  |  |  |  |  |  | 20 | Programming exercises | 2 weeks, released in Week 2, due in Week 4 |
| Assignment 2 |  | X | X | X | X |  |  |  |  |  |  | Programming exercises | 2 weeks, released in Week 5, due in Week 7 |
| Assignment 3 |  |  |  |  |  |  |  |  |  |  |  | Programming exercises | 2 weeks, released in Week 10, due in Week 12 |
| Coursework Exam 1 | X | X | X |  | X |  |  | X | X |  | 15 | Short answer theory questions, programming code exercises, problem-solving tasks, code analysis | 1 hour, Held in Week 5 |
| Coursework Exam 2 | X | X | X |  | X |  | X | X | X | X | 15 | Short answer theory questions, programming code exercises, problem-solving tasks, code analysis | 1 hour, Held in Week 10 |
| Final Examination | X | X | X | X | X |  | X | X | X | X | 50 | Short answer theory questions, programming code exercises, problem-solving tasks, code analysis | 2 hours, written |
| Total % |  | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Benjamin J. Evans and, David Flanagan (2019). Java in a Nutshell: A Desktop Quick Reference. O'Reilly Media; 7th Edition ,,

Kishori Sharan and Peter Späth (2022). Learn JavaFX 17: Building User Experience and Interfaces with Java . Apress; 2nd Edition ,,

BlueJ IDE: https://www.bluej.org/,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Introduction to Java, Classes and Objects | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 2. | Encapsulation, Information Hiding, Classes and Methods | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 3. | Object Relationships and Equality | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs | Assignment 1 | Due in Week 4 |
| 4. | Inheritance, Polymorphism | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 5. | Polymorphism, Abstract Classes | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs | CW Exam 1 | Week 5 |
| 6. | Interfaces and Abstractions | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 7. | Graphical User Interfaces | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs | Assignment 2 | Due in Week 7 |
| 8. | Event Driven Programming | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 9. | Design Fundamentals | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 10. | Container Classes | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs | CW Exam 2 | Week 10 |
| 11. | Container Classes | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs |  |  |
| 12. | Packages, Build Environments | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Lectures, Labs | Assignment 3 | Due in Week 12 |
| 13. | Course Review(No introduction of new subject matter) | Lecture Notes on myElearning, selected chapters/sections from recommended texts, online tutorials/guides | Review |  |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 2604 Operating Systems**

**Semester and Level: Semester 2 Level 2**

**Pre-requisites: COMP 1600**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course looks at the inner workings of operating systems such as Windows, Ubuntu, and Mac OS X, both from a theoretical algorithmic point of view as well as a practical system programming point of view. The student will be introduced to the fundamental algorithms that support the existence of contemporary operating systems. Topics include the important areas of processes, threads, and CPU management, main and virtual memory management, file systems, disk scheduling algorithms, protection and security. This course will be delivered using a combination of lectures, blended learning, case studies, labs and various online resources. Assignments will take the form of written examinations, lab examinations, algorithm implementation and online assignments.

**2. Rationale**

An operating system acts as the interface between software applications and computer hardware and therefore is an essential course to Computer Science. This course presents important topics of operating systems such as structures of computer hardware and OSs, processes, threads, concurrency, CPU scheduling algorithms, primary and virtual memory, page replacement algorithms, file systems, disk scheduling algorithms, protection, and security. The course prepares students for other courses such as Computer Networks and Distributed Systems.

**3. Course Aims**

Operating Systems is to provide a user-friendly and convenient environment. Students are expected to learn and grasp the concepts of Operating Systems, as well as get expertise in how they operate as an intermediate between hardware and users, allowing users to access and utilise other resources more easily to manage a computer system's resources.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Describe how an operating system manages multiple processes on a single-CPU system

2. Analyze the procedure involved in context switching

3. Explain the advantages and disadvantages of inter-process communication (shared memory and message passing)

4. Analyze available solutions for how deadlocks can arise in a system

5. Examine virtual memory management systems using standard and inverted page tables

6. Examine a variety of page replacement algorithms

7. Describe the structure of a hard disk and the basic file system formats

8. Describe how access to resources is controlled by the operating system

9. Write programs to create, control and destroy processes and threads

10. Write programs that use semaphores for critical region control, as well as for synchronization

11. Write programs for inter-process communication (IPC) using shared memory and message passing on UNIX systems, Win32 API, and Java API

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Introduction to Operating Systems   
2. Operating System Structures   
3. Process Management   
4. Thread Management   
5. CPU Scheduling   
6. Process Synchronization (Concurrency)   
7. Main Memory Management   
8. Virtual Memory Management   
9. Storage Management and Disk Scheduling   
10. File System   
11. Security and Protection

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered twice weekly. Recordings available offline post-lecture. Interactive lectures and Lab demonstrations. |
| Online Activities (myElearning) | Polls, Whiteboard |
| Laboratory Work | Programming exercises, short-answer questions, research questions. |
| Flipped Classroom | Video tutorials, Reading assignments |
| Inquiry-based Learning | Online lab activities, Virutal labs, demonstrations, discussions. |
| Problem-based Learning | Programming exercises, case studies, discussions. |
| Groupwork |  |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

This course will be converted to 100% coursework. The assessment components are as follows: 3 Assignments 10% each 1 Coursework Exam 20% 1 Final Exam 50% First assignment includes Programming and short answer questions. Second and third assignments include Programming, alogrithm implementation, short answer questions and report writing. Course work and final exam includes short answer questions, program writing, algorithm implementation and case studies.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Assignment 1 | X | X | X |  |  |  |  | X |  |  |  | 10 | Programming exercises, short-answer questions, research questions. | Take home |
| Assignment 2 | X | X | X |  |  |  |  | X |  | X |  | 10 | Programming exercises, short-answer questions, research questions, algorithm implementation | Take home |
| Assignment 3 |  |  |  |  |  |  |  |  |  |  |  | 10 | Programming exercises, short-answer questions, algorithm implementation and report writing | Take home |
| Course Work Exam | X | X | X | X |  |  |  | X | X |  |  | 20 | Programming exercises, short-answer questions, case studies | 2 hours |
| Final Exam | X | X | X | X | X | X | X | X | X | X | X | 50 | Programming exercises, short-answer questions, algorithm implementation and case studies | 3 hours |
| Total % |  | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Abraham Silberschatz, Peter B. Galvin, Greg Gagne. 2018. Operating System Concepts, 10th Ed, Wiley. ISBN: 9781119320913.,William Stallings. 2018. Operating Systems: Internals and Design Principles, 9th Ed, Pearson. ISBN: 9780134670959.,Andrew S. Tanenbaum, Herbert Bos. 2015. Modern Operating Systems, 4th Ed, Pearson. ISBN: 9780133591620.

,,

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Introduction to Operating Systems | Ch 1 | Lectures, Labs |  |  |
| 2. | Operating System Structures | Ch 2 | Lectures, Labs |  |  |
| 3. | Processes | Ch 3 | Lectures, Labs |  |  |
| 4. | Threads | Ch 4 | Lectures, Labs | Assignment 1 | WK4 |
| 5. | CPU Scheduling | Ch 5 | Lectures, Labs |  |  |
| 6. | Process Synchronization | Ch 6, 7 & 8 | Lectures, Labs | Assignment 2 | WK6 |
| 7. | Main Memory | Ch 9 | Lectures, Labs |  |  |
| 8. | Virutal Memory | Ch 10 | Lectures, Labs | Course Work Exam | WK8 |
| 9. | Mass-Storage Structure | Ch 11 | Lectures, Labs |  |  |
| 10. | File System | Ch 13 & 14 | Lectures, Labs |  |  |
| 11. | Security | Ch 16 | Lectures, Labs | Assignment 3 | WK11 |
| 12. | Protection | Ch 17 | Lectures, Labs |  |  |
| 13. | Course Review(No introduction of new subject matter) |  |  |  |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 2606 Software Engineering I**

**Semester and Level: Semester 2 Level 2**

**Pre-requisites: COMP 1603**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

The specification, development, management, and evolution of software systems make up the discipline of software engineering. In this course, students apply methods and tools to develop software designs and specifications. The course focuses on universal techniques for developing large-scale systems rather than individual algorithms. In order to build good business systems, it is particularly important that the student place a great deal of emphasis in exploring the different process models and the topics covering requirements analysis and system specification, system architecture and design, verification and validation and system evolution. During the course, students will participate in a real problem solving/software development project which will expose them to the processes, tools and techniques of professional product-quality software development.

**2. Rationale**

Software engineering is a practical exercise that requires understanding how large-scale system development differs from small-scale programming and algorithm design. The course exposes students to issues related to the design, development and management of software products. It provides the foundation for making informed decisions about what methodology and tools to use in order to create robust and well-tested systems.

**3. Course Aims**

The course aims to provide the foundation for making informed decisions regarding methodology, requirements, design, development, testing and management of software products.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Illustrate how software can interact with and participate in various systems including information management, embedded, process control and communication systems.

2. Develop project proposals that include comprehensive requirements based upon organization, client and/or system needs.

3. Describe how programming in the large differs from individual efforts with respect to understanding a large code base, code reading, understanding builds, and understanding context of changes.

4. Develop comprehensive requirements and project proposals based on organization, client and/or system needs.

5. Document the differences between software process models including the advantages and disadvantages of each one. (e.g., waterfall, incremental, agile)

6. Describe how general software engineering methodologies are applied¬¬ in specific domains, with a particular focus on dependable systems.

7. Construct structural and behavioural models of a simple software using system design principles and design paradigms (such as structured design)

8. Derive the necessary trade-offs in developing fit for purpose systems through the application of evaluation metrics.

9. Develop strategies to manage identified risks, manage and motivate teams, roles motivation and conflict.

10. Demonstrate inter-personal skills, teamwork, and effective use of appropriate technology associated with the field of computer studies.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Requirements Engineering (elicitation, analysis and specifications, user stories, functional and non -functional requirements)  
2. Software Process Models (waterfall, incremental, agile)  
3. System Modeling (Context, Interaction, Structural, Behavioural)  
4. Object Oriented Software Design and UML   
5. Product and Project Management (Cost Estimation, Risk Management)  
6. Testing, verification and validation (Test-Driven Development (TDD), Test plans, test strategies)

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered either online or face-to-face twice weekly |
| Online Activities (myElearning) | Weekly activities : exercises based on case studies |
| Laboratory Work |  |
| Flipped Classroom | Early release of lecture content, tutorial exercises with extra activities for discussion and research |
| Inquiry-based Learning | Interactive model /design of scenarios during tutorial session |
| Problem-based Learning | Questions based on case studies. |
| Groupwork | Students collaborate in groups of 3 - 4 on problem based on a real world problem |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 1 hour | 1 hour |
| Total: | 13 | 3 hours | 3 hours |

**8. Course Assessments Description**

In this course, assessments are designed to test student knowledge through a combination of theoretical/analytical exercises and a project There are four coursework assessments in this course worth 50% collectively: 2 assignments and 1 written coursework exams based on theoretical concepts and a group project designing a system. The remaining 50% is assessed in a final 2 hour written examination.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Assignment 1 | X | X | X | X | X |  |  |  |  |  | 5 | Short answer theory questions based on a case study | 2 weeks, released in week 4, due in week 6 |
| Assignment 2 |  |  |  |  | X | X | X |  |  |  | 5 | Short answer theory questions based on a case study | 2 weeks, released in week 7, due in week 9 |
| Course Work Examination |  |  |  |  |  |  |  |  |  |  | 20 | Short answer theory questions. | 2 hours held in week 11 |
| Project | X | X | X | X | X | X | X | X | X | X | 20 | Develop the concept of a system from requirements gathering, system modeling, chosing an appropriate architecture, develop test plans and consider risks involved in developing the system. | Given in week 1 due week 12 |
| Final Examination | X | X | X | X | X | X | X | X | X | X | 50 | Short answer theory questions. | 2 hours, written |
| Total % |  | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Sommerville Ian A., Software Engineering 10th edition (2015), Pearson, ISBN -10: 0133943038; ISBN-13: 978-0133943030,,

,,

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Introduction to Software Engineering | Sommerville, Chapter 1 | Lecture, tutorial | Project Given | Due in Week 12 |
| 2. | Software Process Models | Sommerville, Chapter 2 | Lecture, tutorial |  |  |
| 3. | Agile Method | Sommerville, Chapter 3 | Lecture, tutorial |  |  |
| 4. | Requirements Engineering | Sommerville, Chapter 4 | Lecture, tutorial | Assignment #1 | Due in Week 6 |
| 5. | System Modeling : Context Model (Activity Diagram), Interaction Model (Use Case and Sequence Duagrams) | Sommerville, Chapter 5 | Lecture, tutorial |  |  |
| 6. | System Modeling: Structural Model (Class Diagram) , Behavioual Model (State Diagram ) | Sommerville, Chapter 5 | Lecture, tutorial |  |  |
| 7. | Architecural Design: Design decisions, views, patterns, architectures | Sommerville, Chapter 6 | Lecture, tutorial | Assignment #2 | Due in Week 9 |
| 8. | Architecural Design: Design decisions, views, patterns, architectures | Sommerville, Chapter 6 | Lecture, tutorial |  |  |
| 9. | Testing | Sommerville, Chapter 8 | Lecture, tutorial |  |  |
| 10. | Project management | Sommerville, Chapter 22 | Lecture, tutorial | Course Work Exam | Week 11 |
| 11. | Cost Estimation | Sommerville, Chapter 23 | Lecture, tutorial |  |  |
| 12. | Risk management | Sommerville, Chapter 22 | Lecture, tutorial |  |  |
| 13. | Course Review(No introduction of new subject matter) |  |  |  |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: COMP 3607 Object-Oriented Programming II**

**Semester and Level: Semester 2 Level 3**

**Pre-requisites: COMP 2603**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Elective**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course looks at the main tools of modern object-oriented software development. These include design-support techniques and tools (principally design patterns), programming-support, debugging and testing tools (principally IDEs). The course has a strong emphasis on object-oriented design, programming, refactoring and testings. Design pattern represents a best practice solution to a software problem in a specific context and the course examines the rationale and benefits of using patterns for such cases in relation to code smells and refactoring techniques. Numerous problems will be studied to investigate the implementation of good design, implementation and testing of object oriented programs and systems.

**2. Rationale**

Computer programmers and software engineers must be able to use a variety of programs and systems for designing solutions to common information technology issues. The object-oriented programming paradigm has made it easier to handle software development involving complex tasks since it easily facilitates the decomposition of problems into modular entities. The course will allow students to practice advance concepts in object-oriented design. This course will help motivated students to be primary contributors to any small to mid-sized commercial or open-source software project.

**3. Course Aims**

COMP 3607 aims to elevate the practical object-oriented programming skills learned in COMP2603,Object-oriented Programming 1, in the context of production quality code projects and software systems. The course aims to expose students to best practices in modelling, problem-solving, refactoring and testing using well-known tools and techniques.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Analyze modelling problems using object-oriented design principles

2. Design object-oriented solutions using UML and software tools

3. Discuss the principles of good object-oriented design

4. Explain the reasoning for object-oriented design principles.

5. Use design patterns to facilitate good object-oriented design and implementation

6. Apply knowledge of design patterns to solve common programming problems

7. Construct high level class diagrams in UML for object-oriented software solutions.

8. Evaluate consequences of applying design patterns to overall software quality of a system

9. Refactor object-oriented programs according to design principles and best practices

10. Test object-oriented programs with automated unit tests

11. Manage object-oriented projects using distributed version control tools

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1.Algorithms and Design  
1.1 Fundamental design concepts and principles  
2.Basic Type Systems  
2.1 Type safety and errors caused by using values inconsistentlygiventheir intended types  
2.2 Goals and limitations of static typing  
2.3 Generic types (parametric polymorphism)  
3.Data Modeling  
3.1 Conceptual models (e.g., entity-relationship, UML diagrams)  
3.2 Object-oriented models(cross-reference PL/Object-Oriented Programming)  
3.3. Object Persistence using relational databases and other forms  
4.Development Methods  
4.1 Program comprehension  
4.2 Program correctness  
4.3 Simple refactoring  
4.4 Modern programming environments  
4.5 Debugging strategies  
4.6 Documentation and program style  
5.Introduction to Modeling and Simulation  
5.1 Models as abstractions of situations  
6.Object-Oriented Programming  
6.1 Object-oriented design  
6.2 Subclasses, inheritance, and method overriding  
6.3 Subtyping (cross-reference PL/Type Systems)  
7.Programming Interactive Systems  
7.1 Software Architecture Patterns, e.g.,Model-View controller; command objects, online, offline (cross reference PL/Event Drivenand Reactive Programming, where MVC is used in the context of event-driven programming)  
7.2 Modern GUI libraries (e.g. iOS, Android, JavaFX) GUIbuilders and UI programming environments (cross- reference PBD/Mobile Platforms)  
8.Requirements Engineering  
8.1 Describing functional requirements using, for example, use cases or users stories  
8.2 Unified Modelling Language (UML) - Class, Sequence and Use-Case Diagrams  
9.Software Construction  
9.1 Coding standards  
9.1 Model View Controller (MVC) architecture  
10.Software Design  
10.1 System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion, re-use of standard structures  
10.2 Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented  
10.3 Structural and behavioral models of software designs  
10.4 Design patterns  
10.4.1. Strategy, Observer, Factory, Singleton, FlyWeight, Command, Adapter, Facade, Template Method, Iterator, Composite, State, Proxy and Mediator  
10.5 Refactoring designs using design patterns  
10.5.1 Code Smells  
10.6 Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling,  
10.7 SOLID design principles  
11.Software Verification and Validation  
11.1 Test-driven development  
11.2 Object-oriented testing; systems testing  
11.3 Concepts of Code Refactoring and testing  
12.Tools and Environments  
12.1 Testing tools including static and dynamic analysis tools  
12.2 Programming environments that automate parts of program construction processes (e.g., automated builds)

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered either online or face-to-face twice weekly |
| Online Activities (myElearning) | Weekly activities: short quiz, design task/programming task/ forum posting task |
| Flipped Classroom | Early release of lecture content, lab exercises with extra activities for debate/discussion/exploration |
| Inquiry-based Learning | Interactive code writing/ demonstrations / problem solving during lab sessions |
| Problem-based Learning | Exercises/activities/tasks from lectures, labs and worksheets |
| Group work/ Collaborative | Project work, presentations, demonstration videos |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

In this course, assessments are designed to test student knowledge and practical skills through a combination of theoretical design exercises and practical software-related tasks. In addition, students will be assessed through a group project which requires production of a working software solution to an assigned problem, managed using version control and tested using automated techniques. There are 5 assessments in this course: 2 individual assignments, 2 mixed theoretical and practical coursework exams and 1 group project (approximately 3-4 members). This course has 100% coursework and no final examination.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Assignment 1 | X | X | X |  |  |  |  |  |  |  |  | 10 | Programming and/or UML design exercises using Design Patterns | 2 weeks, released in Week 2, due in Week 4 |
| Assignment 2 |  |  |  | X | X | X | X |  |  |  |  | 10 | Programming and/or UML design exercises using Design Patterns | 2 weeks, released in Week 5, due in Week 7 |
| Coursework Exam 1 |  |  |  |  |  |  |  |  |  |  |  | 25 | Short answer questions, programming code exercises, problem-solving tasks, code analysis | 2 hours, Held in Week 5 |
| Coursework Exam 2 |  |  |  |  | X | X | X | X | X |  |  | 25 | Short answer questions, programming code exercises, problem-solving tasks, code analysis | 2 hours, Held in Week 9 |
| Group project | X | X | X | X | X | X | X | X | X | X | X | 30 | Coding project, presentation, documentation, peer-review, code demonstration | Released in Week 3, code base due in Week 11, presentation due in Week 12 |
| Total % |  | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Gamma et al. (1994) Design Patterns: Elements of Reusable Object-Oriented Software . Addison-Wesley Professional; (seminal text),Martin Fowler (2018). Refactoring: Improving the Design of Existing Code (2nd Edition), Addison-Wesley Professional; 2nd edition,Craig Larman (2008). Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development, Dorling Kindersley Pvt Ltd

Robert C. Martin (2018) Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition ,Benjamin J. Evans and, David Flanagan (2019). Java in a Nutshell: A Desktop Quick Reference. : O'Reilly Media; 7th Edition ,

StarUML: http://staruml.io/,Visual Studio Code: https://code.visualstudio.com/

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Introduction to Object-Oriented Analysis and Design, | Larman: Chpt 1, 6 | Lectures, Labs |  |  |
| 2. | UML - Class, Sequence and Use-Case Diagrams | Larman: Chpt (parts) 1, 9, 10, 15, 16 | Lectures, Labs |  |  |
| 3. | SOLID Design Principles | Resources on myElearning | Lectures, Labs | Project Group Formation | Week 3 |
| 4. | SOLID Design Principles, Introduction to Refactoring | Resources on myElearning | Lectures, Labs | Assignment 1 | Due in Week 4 |
| 5. | Introduction to Design patterns, Singleton | Gamma: Chpt 1 | Lectures, Labs | CW Exam 1 | Week 5 |
| 6. | Design Patterns - Composite, Command | Gamma: Chpt 4,5 | Lectures, Labs |  |  |
| 7. | Design Patterns (Behavioural): Observer, Template | Gamma: Chpt 5 | Lectures, Labs | Assignment 2 | Due in Week 7 |
| 8. | Refactoring, Code Smells, Strategy & Iterator Design Patterns | Gamma: Chpt 5; Fowler: Chpt 1 | Lectures, Labs |  |  |
| 9. | Design Patterns - Adapter, Code Smells: Couplers; Façade Pattern | Gamma: Chpt 4 | Lectures, Labs | CW Exam 2 | Week 9 |
| 10. | Design Patterns - Mediator, State, Intro to Maven Projects | Gamma: Chpt 5; | Lectures, Labs |  |  |
| 11. | Object Persistence, JPA, JDBC with MySQL | Resources on myElearning | Lectures, Labs | Project code | Due in Week 11 |
| 12. | Project Week | Resources on myElearning | Lectures, Labs | Project presentations | Due in Week 12 |
| 13. | Course Review(No introduction of new subject matter) |  | Review |  |  |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: INFO 2603 Platform Technologies 1**

**Semester and Level: Semester 1 Level 2**

**Pre-requisites: INFO 1600 OR COMP 1600**

**Co-requisites: DONE**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course provides the student with an introductory understanding of the terminology and concepts of operating systems, computer architecture, and computer networking. The technical foundation of operating systems installation, configuration, administration and troubleshooting are introduced to students. The course will be delivered using a combination of face-to-face lectures and interactive hands-on computer lab sessions, along with eLearning activities using various online resources. There are 5 units in this course: Unit 1: Hardware and Computer Architecture, Unit 2: Operating System Concepts, Unit 3: Operating Systems in Practice, Unit 4: System Administration, Unit 5: Network Operating System Concepts. Assessments will take the form of written examinations, practical lab examinations and a group project. This course has 100% coursework and no final written exam.

**2. Rationale**

A computer platform generally refers to the operating system and computer hardware only. Operating systems are central to all computing activities and act as intermediaries between computer users and a computers hardware and software. Computer platforms however conform to a set of standards that enable software developers to produce software applications for specific platforms. This course therefore bridges the knowledge gap between how computer platforms work with all other software systems and application code to produce a working environment for users. It merges theoretical content with practical, real-world application of computer platforms in business settings. It also prepares students for further courses on computer systems such as computer networks.

**3. Course Aims**

INFO 2603 aims to develop practical operating system administration, management and troubleshooting skills in undergraduate students while promoting an understanding of the theory behind these operations relevant to computer architecture and networking concepts. This addresses the need for students to be able to configure and manipulate basic computer systems for a variety of platforms with a clear understanding of the implications of these tasks for business setting

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Examine the basic hardware, computer architectures and machine organisations used by modern operating systems.

2. Discuss operating system concepts, design objectives and principles using abstractions, layers and APIs

3. Analyse flavours of operating systems and evaluate features that meet the needs of a business in different scenarios.

4. Design an installation strategy for operating systems, the features and software to ensure the most economical use of hardware resources and time using a hypervisor.

5. Configure an operating system, its settings and features so that it can perform specific tasks for a user using virtual machines.

6. Diagnose errors and problems with computers and the network to ensure the systems are available at all times.

7. Perform administration of an operating system allowing ongoing user access to the required resources on a network.

8. Utilise command line environments to manage file and directories, and perform administrative tasks in order to increase administrative performance.

9. Analyse how hardware developments have led to changes in the priorities for the design and the management of file systems and data storage

10. Conduct research necessary to successfully complete an assigned project on a cutting-edge topic area in Computer Science/Information Technology as applied to a business

**5. Course Content/Syllabus**

The following main topics are covered in this course:

UNIT 1 - Hardware Review   
1.Assembly Level Machine Organization  
1.1 Basic organization of the von Neumann machine  
1.2 Shared memory multiprocessors/multicore organization  
1.3 Introduction to SIMD vs MIMD and the Flynn Taxonomy  
2. Resource Allocation and Scheduling  
2.1 Kinds of resources (e.g.,processor share, memory, disk, net bandwidth)  
3. DigitalLogic andDigitalSystems  
3.1. Overview and history of computer architecture  
3.2. Review of major computer organisation models: Von Neumann, Harvard.  
3.3. Modern Processor Architectures: RISC, CISC  
3.4. Instruction Set Architectures x86, x86\_64, IA64, ARM, SPARC.  
3.5. Computer System Components, I/O  
3.6. Bus Interconnection, width, pipelining.  
  
UNIT 2 - Operating Systems Concepts  
4.Overview of Operating Systems  
4.1 Role and purpose of the operating system  
4.2 Functionality of a typical operating system  
4.3 Mechanisms to support client-server models, hand-held devices  
4.4 Design issues (efficiency, robustness, flexibility, portability, security, compatibility)  
4.5 Categories of Operating Systems  
4.6 Operating System Managers  
4.7 Kernel vs User Mode  
  
UNIT 3 - Operating Systems Management  
5.Operating System Principles  
5.1 Abstractions, processes, and resources  
5.2. Process Management: Multithreading, Context Switching  
5.3. Scheduling, Race Condition  
5.4. Hardware-Software Interface, POSIX standard  
5.5. Concrete examples of Operating Systems: UNIX, Linux, Windows  
5.6. Design issues (efficiency, robustness, flexibility, portability, security, compatibility)  
6.File Systems  
6.1 Files: data, metadata, operations, organization, buffering, sequential, nonsequential  
6.2 Directories: contents and structure  
6.3 File systems: partitioning, mount/unmount, virtual file systems  
7.Memory Management  
7.1 Review of physical memory and memory management hardware  
7.2. Main memory organisation and operation  
7.3. Cache memories (address mapping, block size, replacement and store policy)  
8.Fault Tolerance  
8.1 Fundamental concepts: reliable and available systems (cross-reference SF/Reliability through Redundancy)  
9.Device Management  
9.1 Characteristics of serial and parallel devices  
10. AR/Interfacing and Communication  
10.1 I/O Fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O  
10.2 Interrupt structures and acknowledgement  
10.3 External storage, physical organisation and drives  
10.4 Buses: bus protocols, arbitration, direct-memory access (DMA)  
10.5 Introduction to networks, multimedia support  
10.6 RAID architectures  
11.Reliability through Redundancy  
11.1 Redundancy throughcheck and retry  
11.2 Redundancy through redundant encoding (error correcting codes, CRC, FEC)  
11.3 Duplication/mirroring/replicas  
11.4 Other approaches to fault tolerance and availability  
  
UNIT 4 - VIRTUALISATION  
12.Virtual Machines  
12.1 Types of virtualization (includingHardware/Software, OS, Server, Service, Network)  
12.2 Paging and virtual memory  
12.3 Virtual file systems  
13.Virtualization and Isolation  
13.1 Levels of indirection, illustrated by virtual memory for managing physical memory resources  
13.2 Methods for implementing virtual memory and virtual machines  
  
UNIT 5 - DOMAIN-SPECIFIC ARCHITECTURES AND PLATFORMS  
14. Multiprocessing and Alternative Architectures  
14.1 Power Law  
14.2 SIMD and MIMD instruction sets and architectures  
14.3 Interconnection networks   
14.4 Shared multiprocessor memory systems and memory consistency  
15 AR/Performance Enhancements  
15.1 Superscalar architecture  
15.2 Vector processors and GPUs  
15.3 Alternative architectures, special-purpose processors: personal, warehouse, neural, machine learning architectures

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered twice weekly. Recordings available offline post-lecture. |
| Online Activities (myElearning) | Forum activities, Quizzes, Polls |
| Laboratory Work | Short-answer questions, research questions, practical IT activities |
| Flipped Classroom | Video Tutorials, Reading Assignments, Worksheets |
| Inquiry-based Learning | Online lab activities, virtual labs, demonstrations, discussions |
| Problem-based Learning | Exercises, troubleshooting/configuration activities |
| Groupwork | Project, written research article, team video, animation, peer-review activities |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

"In this course, assessments are designed to test student knowledge and practical skills through a combination of written and practical exams. In addition, students will be assessed through a group presentation which requires research and preparation of strategies to address a particular topic or problem in a given domain. There are five assessments in this course: 1 theoretical exam, 2 mixed practical and theoretical exams, 1 fully practical exam, and 1 group project. As mentioned before, this course is 100% coursework. "

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Assignment 1 | X | X | X |  |  |  |  |  |  |  | 7 | Problems & Short Answer Questions | 2 weeks, released in Week 2, due in Week 4 |
| Assignment 2 |  |  |  | X | X | X | X |  |  |  | 8 | Troubleshooting, Problem Solving Tasks | 2 weeks, released in Week 5, due in Week 7 |
| Course work Exam 1 |  |  |  |  |  |  |  |  |  |  | 25 | Scenario-based Problems & Practical Exercises | 2 hours, Held in Week 5 |
| Course work Exam 2 |  |  |  |  | X | X | X | X | X |  | 25 | Scenario-based Problems & Practical Exercises | 2 hours, Held in Week 9 |
| Project | X | X | X |  |  |  |  |  | X | X | 35 | Research Article, Demo, Presentation | Released in Week 3, Demo video and article due in Week 11, presentation due in Week 12 |
| Total % |  | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

,,

,,

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Units 1-2 | Chapters 1, 3, 4 (Ledin)Chapter 1(McHoe & Flynn) | Lectures, Labs, Worksheets | x | x |
| 2. | Units 1-2 | "Chapters 2,3,4,7,8 (McHoe & Flynn)Chapters 4, 7,10,11 (Ledin)" | Lectures, Labs, Worksheets | Assignment 1 Released | x |
| 3. | Unit 3 | "Chapters 13, 15 (McHoe & Flynn)Chapters 4, 7 (Ledin)" | Lectures, Labs, Worksheets | Project Description Released | x |
| 4. | Unit 3 | "Chapters 13, 15 (McHoe & Flynn)Chapters 5, 8 (Ledin)" | Lectures, Labs, Worksheets | x | Assignment 1 Due |
| 5. | Revision /Unit 4 | See Weeks 1-5 | Review Lecture, Written Exam | Coursework Exam 1 | Week 5 |
| 6. | Unit 4 | "Chapters 13, 15 (McHoe & Flynn)Chapters 6,12 (Ledin)" | Lectures, Labs, Worksheets | x | x |
| 7. | Unit 4 | "Chapter 12, (Ledin)Chapter 14(McHoe & Flynn)" | Lectures, Labs, Worksheets | x | Assignment 2 Due |
| 8. | Unit 4 | Chapters 14 (McHoe & Flynn) | Lectures, Labs, Worksheets | x | x |
| 9. | Revision /Unit 5 | See Weeks 1-9 | Review Lecture, Written Exam | Coursework Exam 2 | Week 9 |
| 10. | Unit 5 | "Chapter 9 (McHoe & Flynn)Chapters 13, 14 (Ledin)" | Lectures, Labs, Worksheets | x | x |
| 11. | Unit 5 | "Chapter 9 (McHoe & Flynn)Chapters 13, 14 (Ledin)" | Lectures, Labs, Worksheets | x | x |
| 12. | Course Review (No introduction of new subject matter) / Project Presentations | See resource section on myElearning course page | Presentation, Demo, Article | Group Presentation | Week 12 |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: INFO 3602 Web Programming and Technologies II**

**Semester and Level: Semester 2 Level 3**

**Pre-requisites: INFO 2602**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Core**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

This course focuses on the development of web services and the evaluation and utilization of software tools to provide solutions to common business problems within the market place. Students are exposed to technologies involved in the development of web services and using industry relevant tools for designing, developing and managing web systems. The course will be delivered using a combination of interactive lectures, eLearning, case studies and online resources. Assignments will take the form of lab examinations, group projects and presentations.

**2. Rationale**

This course provides students with the tools and techniques needed to operate effectively within the ICT and wider industry. The course provides an environment to develop practical skills and techniques used in the development and maintenance of web systems within enterprise environments.

**3. Course Aims**

INFO3602 aims to develop practical web programming skills in undergraduate students while promoting an understanding of the theoretical concepts and design considerations behind these systems. The course also aims to expose students to the development tools and programming APIs commonly used in the field.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Install OS-level, native web server and database on a local development environment

2. Write programs using a contemporary, server-side programming language

3. Manage the execution and administration of server-side programs and sites

4. Customise themes and templates for dynamic sites hosted on a Content Management System

5. Control front-end features (appearance, menus, content, navigation) from back-end files

6. Construct custom queries to display and control front-end content

7. Import/export site files, database, config files across development environments

8. Apply permissions, restrictions and authentication for users based on roles

9. Define web service endpoints for headless queries of CMS sites.

10. Deploy sites from local to production/live servers on the web using automated workflows

11. Manage web projects and documentation using distributed version control tools

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1.Web Platforms  
1.1 Web programming languages (e.g., HTML5, Java Script, PHP, CSS)   
1.2 Web platform constraints   
1.3 Software as a Service (SaaS)  
1.4 Web standards  
2. Platform Based Developemnt  
2.1 Programming via platform-specific APIs  
2.2 Content Management System (CMS) software   
3. Information Management Concepts  
3.1 Information systems as socio-technical systems  
3.2 Information capture and representation  
3.3 Declarative and navigational queries, use of links  
3.4 Supporting human needs: searching, retrieving, linking, browsing, navigating  
3.5 Analysis and indexing  
4.Database Systems  
4.1 Use of a declarative query language  
5.Information Storage and Retrieval  
5.1 Faceted search (e.g., using citations, keywords, classification schemes)  
5.2 Naming,repositories, archives  
6.Data, Information and Knowledge  
6.1 Content management models, frameworks, systems, design methods  
6.2 Digital representations of content including numbers, text, images (e.g., raster and vector), video (e.g.,QuickTime, MPEG2, MPEG4), audio (e.g., written score, MIDI, sampled digitized sound track) and animations; complex/composite/aggregate objects; FRB  
6.3 Content structure / management,  
6.4 Evaluate the requirements for data retention and processing for an application.  
7.Programming Interactive Systems  
7.1 Choosing interaction styles and interaction techniques  
8.Data, Information and Knowledge  
8.1 Widget classes and libraries  
8.2 Data-driven applications (database-backed web pages)  
8.3 Design for resource-constrained devices (e.g. small, mobile devices)  
9.Web Security  
9.1 Session management, authentication  
9.1 Server-side security tools, plugins  
10. Privacy and Civil Liberties  
10.1 Privacy implications of widespread data collection for transactional databases, data warehouses,  
10.2 Technology-based solutions for privacy protection (cross-reference IAS/Threats and Attacks/attacks on  
10.3 Assess the quality, accuracy, and timeliness of data or meta-data and its utility for creating information.

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Live lectures delivered twice weekly ; Interactive lectures and lab demonstrations |
| Online Activities (myElearning) | Forum activities, Quizzes, Wiki, Surveys, Polls |
| Problem-based Learning | Practical lab sessions and activities tied to teaching content |
| Flipped Classroom | Video Tutorials, Reading Assignments, Worksheets |
| Collaborative Learning | Groupwork, collaborative activities, |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

In this course, assessments are designed to test student knowledge and problem solving skills through a combination of theoretical/analytical exercises and practical software-related problem-solving tasks. There are five coursework assessments in this course worth 100% collectively: 2 individual assignments based on theoretical and practical concepts, 2 highly practical coursework examinations, and 1 group project.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Assignment 1 | X | X | X |  |  |  |  |  |  |  |  | 5 | Programming problems (PHP) | 10-14 days |
| Assignment 2 |  | X | X |  |  | X |  |  |  |  |  | 5 | Programming problems/Theory Questions (CMS Queries) | 10-14 days |
| Coursework Examination 1 |  |  |  |  |  |  |  |  |  |  |  | 25 | Practical CMS Tasks, Configuration, Customisation, Troubleshooting | 120-150 minutes |
| Coursework Examination 2 |  |  |  |  | X | X | X | X | X |  |  | 25 | Practical CMS Tasks, Configuration, Customisation, Troubleshooting | 120-150 minutes |
| Group Project | X | X | X | X | X | X | X | X | X | X | X | 40 | Website Design, Coded Solution, Live Deployment, Presentation (10 mins), Online documentation (max 5 pages), Peer Reviews, Group size: 3 members | Released in Week 3, Due in Week 12 |
| Total % |  | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Robin Nixon (2021) Learning PHP, MySQL, JavaScript CSS, & HTML5: A Step-by-Step Guide to Creating Dynamic Websites. 6th edition. O'Reilly Media,,

Doug Bierer and Cal Evans. (2021). PHP 8 Programming Tips, Tricks and Best Practices: A practical guide to PHP 8 features, usage changes, and advanced programming techniques , Packt Publishing,Karol Krol (2019) WordPress 5 Complete: Build beautiful and feature-rich websites from scratch, 7th Edition, Packt Publishing,

,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | Introduction to PHP | Lecture notes, resources on myElearning | Lecture, Lab |  |  |
| 2. | Content Management Systems - WordPress | Lecture notes, resources on myElearning | Lecture, Lab |  |  |
| 3. | Installing and Configuring Themes, Pages, Blogs | Lecture notes, resources on myElearning | Lecture, Lab | Group Project released |  |
| 4. | Dynamic Naviagation, Content Hierarchies and Types | Lecture notes, resources on myElearning | Lecture, Lab | Assignment 1 Due | Week 4 |
| 5. | Responsive websites, Blog Content, Archives | Lecture notes, resources on myElearning | Lecture, Lab |  |  |
| 6. | Custom Post Types, Relationships, Queries | Lecture notes, resources on myElearning | Lecture | Coursework Examination 1 | Week 6 |
| 7. | Custom Post Types, Relationships, Queries | Lecture notes, resources on myElearning | Lecture, Lab |  |  |
| 8. | User Management Plugins, Roles, Permissions | Lecture notes, resources on myElearning | Lecture, Lab | Assignment 2 Due | Week 8 |
| 9. | WordPress REST API | Lecture notes, resources on myElearning | Lecture, Lab |  |  |
| 10. | Backup, Restoring and Going Live | Lecture notes, resources on myElearning |  | Coursework Examination 2 | Week 10 |
| 11. | Live Search, User Generated Content | Lecture notes, resources on myElearning | Lecture, Lab |  |  |
| 12. | General Data Protection, Data Management Policies | Lecture notes, resources on myElearning | Lecture, Lab |  | Week 12 |
| 13. | Project Presentations (no new content matter) | Lecture notes, resources on myElearning | Presentations | Group Presentations | Week 13 |

**THE UNIVERSITY OF THE WEST INDIES**

**PROPOSAL FOR REVISED UNDERGRADUATE COURSE**

**Campus and Faculty: St. Augustine Campus, Faculty of Science and Technology School, Department, or Centre: Department of Computing and Information Technology**

**Course Code and Title: INFO 3608 E-Commerce**

**Semester and Level: Semester 2 Level 3**

**Pre-requisites: INFO 2600 OR COMP 2606**

**Co-requisites:**

**Anti-requisites: None**

**Course Type: Elective**

**Credits: 3**

**Projected Enrolment: 100**

**Projected Start Date: January 2020**

**Mode of Delivery: Face-to-Face** ❑ **Blended** ☑ **Online** ❑

**1. Course Description**

The course E-Commerce provides comprehensive coverage of topics from E-Commerce strategy, business models types and categories. It examines the various eCommerce payment systems and surrounding security issues. The course also covers E-commerce technology infrastructure, business concepts, marketing/revenue models, social issues and real world experiences. It explorers the tools and application that goes to creating a E-Commerce solutions.

**2. Rationale**

Firms and institutions of all types and sizes are revaluating their strategies and transforming their mode of operations. Recent global challenges and local pressures have created a need for all organisations to adopt and implement an e-commerce solution that would reap the benefits from the tools / techniques available by the current and progressive technologies. The course challenges participants to explore the realities and implications of e-commerce in a contemporary global business environment. The non-adoption of an e-commerce solution has proven to be the detriment to most, if not all, businesses. E-commerce has therefore been an essential component in all business strategy and operations.

**3. Course Aims**

This course aims to examines the strategies, models and principles of e-commerce. Critical surrounding factors and issues are explored and applied into the process of the design, development, recommendation and implementation of an e-commerce solution.

**4. Course Learning Outcomes**

Upon the successful completion of this course, the student will be able to:

1. Differentiate between e-commerce and e-business, technological building blocks and major current themes in e-commerce.

2. Describe major B2C and B2B business models, concepts and applicable strategies.

3. Articulate how the Web, Internet, web features and services support e-commerce.

4. Explain the process that should be followed in building an e-commerce presence.

5. Identify the key security threats in the e-commerce environment.

6. Discuss major e-commerce payment systems in use today.

7. Identify basic digital commerce marketing and advertising strategies and tools.

8. Contrast traditional online marketing and the new social-mobile-local marketing platforms in terms of key features and the relationships between social, mobile, and local marketing.

9. Examine basic concepts related to privacy and information rights, the practices of e-commerce companies that threaten privacy, and the different methods that can be used to protect online privacy.

10. Describe the major features of the online service sector and business models of on-demand service companies.

11. Interpret major trends in the consumption of media and online content, the major revenue models for digital content delivery, digital rights management, and the concept of media convergence.

12. Analyse how procurement and supply chains relate to B2B e-commerce.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

Course Topics:  
1. Introduction to E-commerce  
1.1.The Difference between E-commerce and E-business  
1.2.Unique Features of E-commerce Technology  
1.3.Types of E-commerce  
1.4.E-commerce: A Brief History  
1.5.Understanding E-commerce: Organizing Themes  
1.6.Academic Disciplines Concerned with E-commerce  
2.E-commerce Business Models and Concepts  
2.1.E-commerce Business Models  
2.2.Major Business-to-Consumer (B2C) Business Models  
2.3.Major Business-to-Business (B2B) Business Models  
2.4.E-commerce Changes Business: Strategy, Structure, and Process  
3.E-commerce Infrastructure: The Internet, Web, and Mobile Platform  
3.1.Internet Backbone, Internet Exchange Points, Tier 3 ISP  
3.2.The Internet and the Web: Features and Services  
3.3.Platforms for Mobile Application Development  
3.4.E-commerce Changes Business: Strategy, Structure, and Process  
4.Building an E-commerce Presence: Websites, Mobile Sites, and Apps  
4.1.Building an E-commerce Presence: A Systematic Approach  
4.2.Determination of Software  
4.3.Determination of Hardware  
4.4.E-Commerce Site Tools  
4.5.Developing a Mobile Website and Building Mobile Applications  
5.E-commerce Security and Payment Systems  
5.1.Security Threats in the E-commerce Environment  
5.2.Technology Solutions  
5.3.Management Policies, Business Procedures, and Public Laws  
5.4.A Security Plan: Management Policies  
5.5.E-commerce Payment Systems  
5.6.Electronic Billing Presentment and Payment  
6.E-commerce Marketing and Advertising Concepts  
6.1.Digital Commerce Marketing and Advertising Strategies and Tools  
6.2.Internet Marketing Technologies  
6.3.Understanding the Costs and Benefits of Online Marketing Communications  
7.Social, Mobile, and Local Marketing  
7.1.Social Marketing  
7.2.Mobile Marketing  
7.3.Local and Location-Based Mobile Marketing  
8.Ethical, Social, and Political Issues in E-commerce  
8.1.Ethical, Social, and Political Issues in E-commerce  
8.2.Privacy and Information Rights  
8.3.Intellectual Property Rights  
8.4.Governance  
8.5.Net Neutrality  
8.6.Antitrust, Monopoly, and Market Competition in the Internet Era  
8.7.Public Safety and Welfare  
9.Online Retail and Services  
9.1.Analyzing the Viability of Online Firms  
9.2.E-commerce in Action: E-tailing Business Models  
9.3.The Service Sector: Offline and Online  
9.4.Online Financial Services  
9.5.Online Travel Services  
9.6.Online Career Services  
10.Online Content and Media  
10.1.Content Market: Entertainment and Media Industry Revenues  
10.2.Online Content: Consumption, Revenue Models, and Revenue  
10.3.The Online Publishing Industry  
10.4.The Online Entertainment Industry  
11.Social Networks, Auctions, and Portals  
11.1.Social Networks and Online Communities  
11.2.Online Auctions  
11.3.E-commerce Portals  
11.4.The Growth and Evolution of Portals  
11.5.Types of Portals: General-Purpose and Vertical Market  
11.6.Portal Business Models  
12.B2B E-commerce: Supply Chain Management and Collaborative Commerce  
12.1.An Overview of B2B E-commerce  
12.2.The Procurement Process and Supply Chains  
12.3.Trends in Supply Chain Management and Collaborative Commerce  
12.4.Net Marketplaces: The Selling Side of B2B  
12.5.Private Industrial Networks

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Interactive Lectures | Online lectures are performed that engage the students to answer impromtu questions to encourage participation and solicit their opinion on current topics related to e-commerce. |
| Online Activities (myElearning) | Online quizzes are provided to assess their knowledge of covered topics. |
| Laboratory Work | Hands-on demonstration on the installation and configuration of an eCommerce solution is conducted.Each session explores the incorporation of a feature or functionality that can be considered in their recommnetation/solution. |
| Flipped Classroom | Students are occasionally asked to demonstrate to other studentshow certain features are installed and configured for used in the development of their own projects. |
| Inquiry-based Learning | This is performed by via "Analysis of Problem and Proposed Solutions" assignment (see details below) |
| Problem-based Learning | Global and local cases are presented in online lectures and relevant solutions are discussed. |
| Groupwork | Students are grouped to work on the design, development and implementation of an e-commerce. This should be properly documented and reported along with a presentation. |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
| Lecture | 13 | 2 hours | 2 hours |
| Computer Science/Information TechnologyTutorial/Labs | 13 | 2 hours | 2 hours |
| Total: | 13 | 4 hours | 4 hours |

**8. Course Assessments Description**

In this course, assessments are designed to test student knowledge and practical skills through a combination of theoretical, analytical and practical exercises and tasks. There are 5 assessments in this course: 2 individual quizzes, 1 group project (approximately 4-5 members), 1 individual research paper and case-based analysis paper. This course has 100% coursework and no final examination.

**9. Course Assessment Type and Course Learning Outcome Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment** | **Learning Outcomes** | | | | | | | | | | | | **Weighting %** | **Assessment**  **Description** | **Duration** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Assessment 1: 10% Quiz | X | X | X | X | X | X |  |  |  |  |  |  | 10 | Twenty (20) MCQ, 10 Short Answers, 1hr, 4th week, synchronous | 1 hour |
| Assessment 2: 10% Quiz |  |  |  |  |  |  | X | X | X | X | X | X | 10 | Twenty (20) MCQ, 10 Short Answers, 1hr, 11th week, synchronous | 1 hour |
| Assessment 3: 30% Group Project |  |  |  |  |  |  |  |  |  |  |  |  | 30 | Design, Develop, Deploy, Document, Present E-Commerce Solution/System. Given in 2nd Week, Due on 12th Week, Group size = 4-5 | 11 week project |
| Assessment 4: 20% Applied Research Paper of Specified list of ECommerce Topics (Individual) | X | X | X | X | X | X | X | X | X | X | X | X | 20 | Given in 2nd Week, Due on 4th Week. Individual research of local business and implemenation of eCommerce solution based on topics discussed in lectures, 5 typed pages equivalent to 2250 words,Based on font: arial, font size: 12, single spaced (http://wordstopages.com) | 2 week |
| Assessment 5: 30% Case Based Analysis (4 Essay Type Questions - on Analysis of Problem and Proposed Solutions) | X | X | X | X | X | X | X | X | X | X | X | X | 30 | Four (4) Essay Type Questions - on Analysis of Problem and Proposed Solutions) (Individual) (Given 5th Week, Due on 9th Week. Answers to each question should be 1 page equivalent to 450 words single spaced, therefore total is 4 typed pages equivalent to 1800 words. Based on font: arial, font size: 12, single spaced (http://wordstopages.com) | 5 week |
| Total % |  | | | | | | | | | | | | 100% |  |  |

**10. Readings/Learning Resources** *(Online and Print)*

*Required/Essential*

Kenneth Laudon (2019) E-commerce 2019: Business, Technology and Society, 15th Edition, Pearson ,,

https://wordpress.org/,Prototyping Tools: https://www.figma.com/,https://wordpress.org/plugins/woocommerce/

XAMPP, Wordpress, WooCommerce,

**11. Staffing Requirements:** 1 Lecturer, 1 Tutor, 1 Marker

**12. Projected additional Cost (if any) for Proposed Undergraduate Course: None**

**13. Collaboration/ Consultation**

Academic staff from the Department of Computing and Information Technology were consulted on the proposal and full support was given for the changes to the curriculum.

**14. All relevant BUS Policies are available at: http://uwi.edu/undergraduatestudies/ default.aspx**

Have you taken these policies into account in the design of this Course? **Yes No \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Academic Staff Member / Contact Person Responsible/Coordinator**

Name: Dr. Wayne Goodridge Telephone: 868-662-2002 ext 85381

Email: wayne.goodridge@sta.uwi.edu

**Campus/Faculty/Department**:

**Date of Recommendation by Faculty Board/APAD:**

**Signature: Dean/Director\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: Department Head**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Checked and endorsed by:** ❑ Campus Bursary ❑ CETL ❑ Library ❑ Bookshop ❑ Faculties on other Campuses ❑ OOL ❑ Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ NB. Attach supporting documents as appendix e.g. CETL. Library, online checklist

**Course Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Topics** | **Required**  **Readings**  **Learning**  **Resources** | **Learning**  **Activities** | **Assessment** | |
| **Name** | **Date** |
| 1. | 1. Introduction to E-commerce | E-commerce 2019: business. technology. society. 15th Edition, Chapter 1 |  |  |  |
| 2. | 2. E-commerce Business Models and Concepts | E-commerce 2019: business. technology. society. 15th Edition, Chapter 2 |  | (Design, Develop, Deploy, Document, Present E-Commerce Solution/System) Given in 2nd Week, Due on 12th Week, Group size = 4-5(Given in 2nd Week, Due on 4th Week) (Individual research of local business and implemenation of eCommerce solution based on topics discussed in lectures, 5 typed pages equivalent to 2250 words,Based on font: arial, font size: 12, single spaced (http://wordstopages.com) |  |
| 3. | 3.E-commerce Infrastructure: The Internet, Web, and Mobile Platform | E-commerce 2019: business. technology. society. 15th Edition, Chapter 3 |  |  |  |
| 4. | 4. Building an E-commerce Presence: Websites, Mobile Sites, and Apps | E-commerce 2019: business. technology. society. 15th Edition, Chapter 4 |  | (20 MCQ, 10 Short Answers, 1hr, 4th week) synchronousIndividual research of local business and implemenation of eCommerce solution based on topics discussed in lectures - DUE |  |
| 5. | 5.E-commerce Security and Payment Systems | E-commerce 2019: business. technology. society. 15th Edition, Chapter 5 |  | (4 Essay Type Questions - on Analysis of Problem and Proposed Solutions) (Individual) (Given 5th Week, Due on 9th Week). Answers to each question should be 1 page equivalent to 450 words single spaced, therefore total is 4 typed pages equivalent to 1800 words. Based on font: arial, font size: 12, single spaced (http://wordstopages.com) |  |
| 6. | 6.E-commerce Marketing and Advertising Concepts | E-commerce 2019: business. technology. society. 15th Edition, Chapter 6 |  |  |  |
| 7. | 7.Social, Mobile, and Local Marketing | E-commerce 2019: business. technology. society. 15th Edition, Chapter 7 |  |  |  |
| 8. | 8. Ethical, Social, and Political Issues in E-commerce | E-commerce 2019: business. technology. society. 15th Edition, Chapter 8 |  |  |  |
| 9. | 9.Online Retail and Services | E-commerce 2019: business. technology. society. 15th Edition, Chapter 9 |  | Essay Type Questions - DUE |  |
| 10. | 10.Online Content and Media | E-commerce 2019: business. technology. society. 15th Edition, Chapter 10 |  |  |  |
| 11. | 11.Social Networks, Auctions, and Portals | E-commerce 2019: business. technology. society. 15th Edition, Chapter 11 |  | (20 MCQ, 10 Short Answers, 1hr, 11th week) synchronous |  |
| 12. | 12.B2B E-commerce: Supply Chain Management and Collaborative Commerce | E-commerce 2019: business. technology. society. 15th Edition, Chapter 12 |  | Design, Develop, Deploy, Document, Present E-Commerce Solution/System - DUE |  |
| 13. | Course Review(No introduction of new subject matter) |  |  |  |  |