**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 programming language as a tool to teach fundamental programming concepts. 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 students to solve problems on computer based systems. It identifies what type of problems can be solved by such systems and which cannot. It guides students on methods of developing structured algorithms. The focus on this course is problem description and presentation using either flowcharting or pseudocode tools. The selected programming language is used as a vehicle to show the basics of programming algorithms.

**3. Course Aims**

**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 and describe uses of primitive data types.

7. Write programs that use primitive data types.

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

9. 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.

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

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

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

13. 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)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 | 11 | 12 | 13 |
| Assingment 1 | X | X | X |  |  |  |  |  |  |  |  |  |  | 5 | Take home problems | 10 days |
| Assingment 2 |  |  | X | X | X |  | X | X |  |  | X |  |  | 6 | Take home problems | 10 days |
| Assingment 3 |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 | Take home problems | 10 days |
| Quiz 1 | X | X | X | X | X |  |  |  |  |  |  |  |  | 1 | Online quiz, multiple choice | 30 min |
| Quiz 2 |  |  | X | X | X | X | X |  |  |  | X |  |  | 1 | Online quiz, multiple choice | 30 min |
| Quiz 3 |  |  |  |  | X | X |  |  | X |  | X |  |  | 1 | Online quiz, multiple choice | 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 |  |  | 20 | Problems & short answer questions | 1.5 hours |
| Final Examinaiton | 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. | CWE1, 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, CWE2 | C Programming - A Beginner's Course: Chapter 8 | Lectures, Tutorials, Worksheets, online videos | Crousework 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 2611 Data Structures**

**Semester and Level: Semester 1 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**

A data structure is a way of storing data in a computer so that it can be used efficiently. Data structures is an important part of the equation; Programs = Algorithms + Data structures. Often a carefully chosen data structure will allow the most efficient algorithm to be used. A well-designed data structure allows a variety of critical operations to be performed, minimizing the use of execution time and memory space. This course covers some fundamental data structuresstacks, queues, linked lists, binary trees, heaps and graphswhich are required for programming the solutions to a wide variety of real-world and theoretical problems.

**2. Rationale**

A critical aspect of solving any problem on a computer is its representation. Knowledge of data structures gives one a better chance of choosing an appropriate representation for a given problem.

**3. Course Aims**

This course aimes to introduce students to fundamental concepts in data structures.This course teaches students the foudations of working with data structures, how infomration can be structured for time and space efficiency, and re-use of code. Mastery of these essenatil topics in Computer Science allow students to design aind implement memory and time effictent solutions specific to the needs of a problem.

**4. Course Learning Outcomes**

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

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

2. Implement common quadratic, O(N log N), and integer sorting algorithms.

3. Implement and solve problems using hash tables, including collision avoidance and resolution.

4. Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing.

5. Explain how tree balance affects the efficiency of various binary search tree operations.

6. Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting.

7. Describe and implement different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees.

8. Describe and implement the heap property, and priority queues with heaps.

9. Implement and solve problems using fundamental graph algorithms, including depth-first, breadth-first search, single-source, all-pairs shortest paths, and at least one minimum spanning tree algorithm.

10. Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system.

11. Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context.

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Sorting  
1.1. Worst case quadratic sorting algorithms (selection, insertion)   
1.2. Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort)  
1.3. Integer sorting algorithms average case O(N)  
2. Sequential and binary search algorithms   
3. Hash tables, including strategies for avoiding and resolving collisions e.g. linear, quadratic, chaining, double hashing  
4. Trees. Properties and traversal strategies  
5. Binary search trees. Common operations on binary search trees such as select min, max, insert, delete, iterate over tree   
6. Heaps. Priority queues.  
7. Graphs and graph algorithms  
7.1. Directed, undirected, and weighted graphs  
7.2. Representations of graphs (e.g., adjacency list, adjacency matrix)  
7.3. Shortest-path algorithms (Dijkstras and Floyds algorithms)  
7.4. Minimum spanning tree (Prims and Kruskals algorithms)  
7.5. Graph isomorphism  
8. Efficiently storing and manipulating matrices with special properties, e.g. symmetric, triangular, band, sparse and others

**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 | Write programs to solve problems. Can make use of online-judges for practice. |
| Groupwork |  |
|  |  |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 | 11 |
| Assingment 2 | X | X | X | X | X |  |  |  |  |  | X | 6 | Take home problems | 10 days |
| Assingment 3 |  |  |  |  | X |  |  | X | X | X | X | 6 | Take home problems | 10 days |
| Quiz 1 |  |  |  |  |  |  |  |  |  |  |  | 1 | Online quiz, multiple choice | 30 min |
| Quiz 2 |  |  |  | X | X | X | X |  |  |  | X | 1 | Online quiz, multiple choice | 30 min |
| Quiz 3 |  |  |  |  |  |  |  | X | X | X | X | 1 | Online quiz, multiple choice | 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 | 20 | Problems & short answer questions | 1.5 hours |
| Final Examinaiton | 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*

Recommended

**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. | 1.2, 1.3 | Data Structures in C - Chapter 6 | Lectures, Tutorials, Worksheets, online videos | Assignment 1 | WK2 |
| 3. | 2, 3 | Data Structures in C - Chapter 5, 8 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 4. | 3, 4 | Data Structures in C - Chapter 5, 8 | Lectures, Tutorials, Worksheets, online videos | Quiz 1 | WK4 |
| 5. | 4, 5 | Data Structures in C - Chapter 5 | Lectures, Tutorials, Worksheets, online videos | Assignment 2 | WK5 |
| 6. | 5, 6 | Data Structures in C - Chapter 5, 6 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 7. | 6, 7.1 | Data Structures in C - Chapter 6, 7 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 8. | CWE1, 7.2 | Data Structures in C - Chapter 7 | Lectures, Tutorials, Worksheets, online videos | Coursework Exam 1 | WK8 |
| 9. | 7.3 | Data Structures in C - Chapter 7 | Lectures, Tutorials, Worksheets, online videos | Quiz 2 | WK9 |
| 10. | 7.4 | Data Structures in C - Chapter 7 | Lectures, Tutorials, Worksheets, online videos | Assignment 3 | WK10 |
| 11. | 7.5, 8 | Data Structures in C - Chapter 7, 9 | Lectures, Tutorials, Worksheets, online videos | Quiz 3 | WK11 |
| 12. | 8, CWE2 | Data Structures in C - Chapter 9 | Lectures, Tutorials, Worksheets, online videos | Crousework 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 3609 Game Programming**

**Semester and Level: Semester 1 Level 3**

**Pre-requisites: COMP 2603 AND 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 Game Programming course will allow students to combine concepts taught in order courses together with the new game programming concepts taught in this course, in order to build games. The students will be introduced to an appropriate 2D programming API (e.g., Java), the game loop, game entities, images, sound, animations, game physics and user input. At the end of the course students will have a good grasp on the concepts of game programming and will be able to produce games for multiple platforms. The course covers the fundamental aspects of images, sounds, animations and sprites and shows how to develop a two-dimensional game using these elements. Mathematics and Physics principles are discussed throughout the course whenever they are pertinent to the topics being presented (e.g. collision detection of sprites).

**2. Rationale**

Game programming involves the application of a wide range of skills and concepts from different areas of Computer Science and other disciplines. These include programming, data structures, computer architecture, operating systems, human computer interaction, and software engineering. Thus, the Game Programming course is intended to give students an introduction to the Computer Science aspects of game programming using an object-oriented programming language.

**3. Course Aims**

This course aims to introduce students to the foundational concepts in Game Programming. This course teaches students how to apply core concepts in computer science with game design, in order to build games. This course allows students to design and implement, single screen, platformer, and top down games from scratch as well with an appropriate game engine.

**4. Course Learning Outcomes**

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

1. Use an appropriate 2D API to draw and colour various shapes.

2. During the execution of a game manipulate identified characteristics of images and sounds

3. Implement user interaction with the keyboard and mouse

4. Create a 2D platform game

5. Implement a specific set of basic game design principles

6. Implement some basic principles of game design

7. Develop by course end at least one game illustrating the design principles, image characteristics and sounds games outlined.

8. Use an appropriate game engine to implement a basic 2D game

**5. Course Content/Syllabus**

The following main topics are covered in this course:

1. Review of Object-oriented design  
1.1 Inheritance   
1.2 Interfaces   
1.3 Inner Classes   
1.4 Using the 2D API Documentation   
2. Introduction to 2D Graphical API   
2.1 Creating and drawing shapes   
2.2 Drawing Text to screen   
3. Basic Game Programming Concepts   
3.1 The basic game loop   
3.2 Drawing and Updating   
3.3 Game Entities   
4. Movement and User Input  
4.1 Events and event handlers   
4.2 Moving a simple 2D shape   
4.3 Getting input from the user  
4.4 Using a reactive framework  
5. Collision detection and response  
5.1 Bounding rectangles   
5.2 Rectangle intersection   
5.3 Screen boundaries   
6. Design Concepts   
6.1 Singleton Design Pattern   
6.2 Composite Design Pattern   
6.3 Hash tables   
6.4 Array Lists   
7. Images and Sprites   
7.1 Loading images   
7.2 The Sprite class   
7.3 Flickering, double buffering, tearing, and page flipping   
7.4 Image Characteristics (Opaque, Transparent, Translucent, etc.)   
7.5 Asset Manager   
8. Animation   
8.1 Animation as a sequence of still images  
8.2 Loading and manipulating Sprite sheets   
8.3 AnimatedSprite class   
9. Sound   
9.1 A Sound API   
9.2 MIDI   
9.3 Loading and playing sound file in Java   
9.4 Sound Effects   
9.5 SoundManager class   
10. Parallax Backgrounds   
10.1 Simulating movement and distance in games   
10.2 ParallaxBackground class   
11. Movement and User Input   
11.1 Managing user input   
11.2 KeyboardInputService class   
12. Design Concepts   
12.1 Object Pool design pattern   
12.2 Game States   
12.3 Managing game screens   
13. 2D Platformer   
13.1 Side Scroller Concept   
13.2 Creating a game camera   
13.3 Tiled Map Editor   
13.4 Loading and displaying tile maps   
13.5 Map collision detection   
14. Simple Game Physics   
14.1 Smooth jump motion and Gravity   
14.2 Projectile motion   
15. Special Topics in Game Programming   
15.1 Implementing games in a game engine  
15.1 Simulating 3D Using 2D   
15.2 Real time strategy games   
15.3 Multi-player games

**6. Teaching Methods**

|  |  |
| --- | --- |
| **Teaching Method** | **Description** |
| Online Activities (myElearning) |  |
| 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 game design concepts to build games |
| Groupwork |  |
|  |  |

**7. Contact and credits hours:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Duration**  **(Number of weeks)** | **Contact Hours**  **(Weekly)** | **Credit Hours**  **(Weekly)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Assignment #2 | X | X | X | X | X | X |  |  | 5 | Take home problems | 14 days |
| Assignment #3 | X | X | X | X | X | X |  |  | 5 | Take home problems | 14 days |
| Assignment #4 |  |  |  |  |  |  |  |  | 6 | Take home problems | 14 days |
| Presentations | X | X | X | X | X | X | X | X | 5 | Present game design and implementation to the class | 20 mins |
| Practical Exam | X | X | X | X | X | X | X | X | 25 | Problems & short answer questions | 1.5 hours |
| Final Examination | 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*

Davison, A., Killer Game Programming in Java. Sebastopol, California: OReilly Media Inc. 2005. Book Web Site: http://fivedots.coe.psu.ac.th/~ad/jg/

Lecture Notes

**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, 3 | Developing Games in Java: Chapter 1, 2 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 2. | 4, 5 | Developing Games in Java: Chapter 2, 3 | Lectures, Tutorials, Worksheets, online videos | Assignment #1 | WK2 |
| 3. | 6, 7 | Developing Games in Java: Chapter 4, 5 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 4. | 8, 9 | Developing Games in Java: Chapter 2, 4 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 5. | 10, 11 | Developing Games in Java: Chapter 3, 5 | Lectures, Tutorials, Worksheets, online videos | Assignment #2 | WK5 |
| 6. | 12 | Developing Games in Java: Chapter 5 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 7. | 13 | Developing Games in Java: Chapter 5 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 8. | 14 | Lecture Notes | Lectures, Tutorials, Worksheets, online videos | Assignment #3 | WK8 |
| 9. | 15.1 | Lecture Notes | Lectures, Tutorials, Worksheets, online videos |  |  |
| 10. | 15.2, 15.3 | Developing Games in Java: Chapter 9 | Lectures, Tutorials, Worksheets, online videos |  |  |
| 11. | 15.4 | Developing Games in Java: Chapter 6 | Lectures, Tutorials, Worksheets, online videos | Assignment #4 | WK11 |
| 12. | Course Review(No introduction of new subject matter) |  | Lectures, Tutorials, Worksheets, online videos | Presentation, Practical Exam | WK12 |
| 13. | Course Review(No introduction of new subject matter) |  |  |  |  |