

#### DEPARTMENT OF COMPUTING

PROGRAMMES

**B.Sc.**

1. **Computer Studies**
2. **Computer Systems Engineering *\*Not being offered 2022/2023\****
3. **Information Technology**
4. **Software Engineering [Mobile Application Technologies] *\*Not being offered 2022/2023\****

**MAJORS**

1. **Computer Science**
2. **Software Engineering**

**MINORS**

1. **Computer Science**
2. **Information Technology**
3. **Software Engineering**

| **Undergraduate Courses Offered by the DEPARTMENT OF COMPUTING** | | | | |
| --- | --- | --- | --- | --- |
| **CODES** | **TITLES** | **CREDITS** | **SEMESTER OFFERED** | **PREREQUISITES** |
| **LEVEL 1** | | | | |
| COMP1126 | Introduction to Computing I | 3 | 1 or 2 | Any one of the following:  CAPE (or A-level) Science subject  ECON1003 OR Teacher’s College Diploma OR Associate Degree in Mathematics or Science or Information Technology |
| COMP1127 | Introduction to Computing II | 3 | 1 or 2 | Any one of the following:  CAPE (or A-level) Science subject  ECON1003 OR Teacher’s College Diploma OR Associate Degree in Mathematics or Science or Information Technology |
| COMP1161 | Object-Oriented Programming | 3 | 1 or 2 | COMP1126 and COMP1127 |
| COMP1210 | Mathematics for Computing | 3 | 1 or 2 | CSEC Mathematics |
| COMP1220 | Computing and Society | 3 | 1 or 2 | None |
| SWEN1007 | Software Engineering Essentials | 3 | 2 | None |
| **LEVEL 2** | | | | |
| COMP2130 | Systems Programming | 3 | 1 | COMP1126, COMP1127 and COMP1161 |
| COMP2140 | Software Engineering | 3 | 1 | COMP1126, COMP1127 and COMP1161 |
| COMP2171 | Object Oriented Design and Implementation | 3 | 2 | COMP2140 |
| COMP2190 | Net-Centric Computing | 3 | 1 | COMP1126, COMP1127, COMP1161, and (COMP1210 or MATH1152)  May not be credited with COMP3150(CS32Q) |
| COMP2201 | Discrete Mathematics for Computer Science | 3 | 1 | COMP1210 or MATH1152 |
| COMP2211 | Analysis of Algorithms | 3 | 2 | COMP1126, COMP1127, COMP1161 and COMP1210 |
| COMP2340 | Computer Systems Organization | 3 | 2 | COMP1126, COMP1127, COMP1161 and COMP1210 |
| COMP2802 | Speech Processing | 3 | 2 | ELET2460, COMP1126 and COMP1127 |
| INFO2100 | Mathematics and Statistics for IT | 3 | 2 | COMP1210 |
| INFO2110 | Data Structures for IT | 3 | 1 | COMP1126, COMP1127 and COMP1161 |
| INFO2180 | Dynamic Web Development 1 | 3 | 1 | COMP1126, COMP1127 and COMP1161 |
| SWEN2165 | Requirements Engineering | 3 | 2 | COMP2140 or SWEN1007 |
| **LEVEL 3** | | | | |
| COMP3092 | An Introduction to Quantum Computing | 3 | 1 | COMP2211, INFO2110 or PHYS2351 |
| COMP3101 | Operating Systems | 3 | 1 | COMP2340 |
| COMP3161 | Database Management Systems | 3 | 2 | COMP1210, COMP1126, COMP1127 and COMP1161 |
| COMP3162 | Data Science Principles | 3 | 2 | (COMP2201 OR INFO2100) AND (COMP2211 OR INFO2110) |
| COMP3191 | Principles of Computer Networking | 3 | 1 | COMP2190 |
| COMP3192 | Implementation of Computer Networks | 3 | 2 | COMP3191 |
| COMP3220 | Principles of Artificial Intelligence | 3 | 1 | COMP2211 and COMP2201 |
| COMP3410 | Introduction to Parallel Computing | 3 | 2 | (COMP2211 or COMP2201) and COMP2340 |
| COMP3652 | Language Processors | 3 | 1 | COMP2211 |
| COMP3702 | Theory of Computation | 3 | 2 | COMP2201 |
| COMP3801 | Real-Time Embedded Systems | 3 | 2 | COMP2340 and COMP2140 |
| COMP3802 | Speech and Language Technology | 3 | 1 | COMP2802 or ELET2210 |
| COMP3901 | Capstone Project | 3 | 2 and 3 | COMP2140, COMP2211, and any 6 credits of Level 2 or 3 Computing code courses |
| COMP3911 | Internship in Computing I | 3 | 1, 2 or 3 | Permission of the Head of Department |
| COMP3912 | Internship in Computing II | 6 | 1, 2 or 3 | Permission of the Head of Department |
| INFO3105 | Computer System Administration | 3 | 1 | COMP2340 and COMP2190 |
| INFO3110 | Information Systems | 3 | 2 | COMP2140 and COMP2190 |
| INFO3155 | Information Assurance and Security | 3 | 2 | COMP2190 and (COMP2201 or INFO2100) |
| INFO3170 | User Interface Design For IT | 3 | 1 | COMP2140 or INFO2180 |
| INFO3180 | Dynamic Web Development II | 3 | 2 | INFO2180 |
| INFO3435 | Ecommerce | 3 | 2 | COMP2140 and INFO2180 |
| SWEN3000 | Application Development for iOS Devices | 3 | 2 | COMP2171 |
| SWEN3001 | Android Application Development I | 3 | 1 | COMP2171 and COMP3161 |
| SWEN3002 | Android Application Development II | 3 | 2 | SWEN3001 |
| SWEN3003 | Web &Mobile Application Development I | 3 | 1 | SWEN1005, COMP2171 and COMP3161 |
| SWEN3004 | Web & Mobile Application Development II | 3 | 2 | SWEN3003 |
| SWEN3120 | Software Architecture | 3 | 1 | COMP2140 and COMP2171 |
| SWEN3130 | Software Project Management | 3 | 1 | COMP2140 |
| SWEN3145 | Software Modelling | 3 | 1 | COMP2140 and COMP2171 |
| SWEN3165 | Software Testing | 3 | 2 | COMP2140 and COMP2171 |
| SWEN3185 | Formal Methods and Software Reliability | 3 | 2 | COMP2201 |
| SWEN3920 | Capstone Project (Software Engineering) | 6 | 1, 2 or 3 | COMP2140, SWEN3130, and SWEN3145 |
| **LEVEL 4** | | | | |
| SWEN4001 | Advanced Database Systems | 3 | 2 | COMP3161 or SWEN2005 |
| SWEN4002 | IT Certification I (Course Shell) | 3 | 1 | None |

##### PROGRAMME DETAILS

|  |  |  |
| --- | --- | --- |
| COMPUTER STUDIES (B.Sc.) | | |
| Introductory  Courses  (Level 1) | **A B.Sc. in Computer Studies requires a total of thirty-six (36) Level 1 credits from:** | |
| COMP1220 | Computing and Society (optional) |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| MATH1141 | Introductory Linear Algebra and Analytic Geometry |
| MATH1142 | Calculus I |
| MATH1151 | Calculus II |
| MATH1152 | Introduction to Formal Mathematics |
| ECON1000 | Principles of Economics I |
| ECON1012 | Principles of Economics II |
| **Either** |  |
| ACCT1005 & ACCT1003 | Financial Accounting &  Introduction to Cost & Management Accounting |
| **OR** |  |
| SOCI1002 & PSYC1002 | Sociology for the Caribbean & Introduction to Industrial/Organizational Psychology |
| Advanced Courses  (Levels 2 and 3) | **A B.Sc. in Computer Studies requires a minimum of thirty-three (33) credits from Computing courses at Levels 2 and 3 and must include:** | |
| COMP2140 | Software Engineering |
| COMP2171 | Object Oriented Design and Implementation |
| COMP2190 | Net-Centric Computing |
| COMP2201 | Discrete Mathematics for Computer Science |
| COMP2211 | Analysis of Algorithms |
| COMP2340 | Computer Organization |
| COMP3101 | Operating Systems |
| COMP3161 | Database Management Systems |
| COMP3220 | Principles of Artificial Intelligence |
| COMP3901 | Capstone Project |
| INFO3110 | Information Systems |
| AND twenty-seven (27) credits from Levels 2 or 3 courses offered by Computing, Mathematics, Economics or Management Studies. | | |

|  |  |  |
| --- | --- | --- |
| COMPUTER SYSTEMS ENGINEERING (B.Sc.)  [Non-UGC Funded] \*Not being Offered 2022/2023\* | | |
| Introductory Courses  (Level 1) | **A B.Sc. in Computer Systems Engineering requires a total of thirty-four (34) Level 1 credits from:** | |
| **Semester 1** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1220 | Computing and Society |
| ECNG1000 | Mathematics for Computing |
| ENGR1000 | Introduction to Engineering |
| MATH1180 | Engineering Mathematics I |
| **Semester 2** | |
| COMP1161 | Object-Oriented Programming |
| ECNG1012 | Electrical Circuits |
| ELET1400 | Introduction to Electronics |
| ELET1405 | Practices in Basic Electronics |
| ELNG1101 | Physics for Engineers |
| Advanced Courses  (Levels 2 and 3) | **A B.Sc. in Computer Systems Engineering requires a minimum of sixty-one (61) credits from Levels 2 and 3 credits and must include:** | |
| **Level 2: Semester 1** | |
| COMP2140 | Software Engineering |
| COMP2190 | Net-Centric Computing |
| COMP2201 | Discrete Mathematics for Computer Science |
| ELET2405 | Practices in Electronics Design I |
| ELET2430 | Digital Circuits and Microprocessors |
| ELET2450 | Embedded Systems |
| **Level 2: Semester 2** | |
| COMP2130 | System Programming |
| COMP2211 | Analysis of Algorithms |
| INFO2180 | Dynamic Web Development I |
| INFO3105 | Computer Systems and Administration |
| MATH2201 | Probability and Statistics for Engineers |
| **Summer Term** | |
| COMP3911 | Internship in Computing I |
| **Level 3: Semester 1** | |
| COMP3101 | Operating Systems |
| COMP3191 | Principle of Computer Networking |
| ECNG3021 | Introduction to Engineering Management and Accounting Systems |
| ELET2460 | Signal and Systems |
| INFO3180 | Dynamic Web Development II |
| **Electives** | |
| ELET3485 | Introduction to Robotics |
| INFO3155 | Information Assurance and Security |
| **Level 3: Semester 2** | |
| COMP3801 | Real Time Embedded Systems |
| COMP3901 | Capstone Project |
| MGMT3136 | New Venture Creation and Entrepreneurship |
| **Electives** | |
| ECNG3016 | Advanced Digital Electronics |
| MATH2230 | Engineering Mathematics |

|  |  |  |
| --- | --- | --- |
| INFORMATION TECHNOLOGY (B.Sc.) | | |
| Introductory Courses  (Level 1) | **A B.Sc. in Information Technology requires a total of fifteen (15) Level 1 credits from:** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| COMP1210 | Mathematics for Computing |
| **Elective** | |
| COMP1220 | Computing and Society |
| Advanced Courses  (Levels 2 and 3) | **A B.Sc. in Information Technology requires a minimum of forty-two (42) credits from Computing Courses at Levels 2 and 3 and must include:** | |
| COMP2140 | Software Engineering |
| COMP2190 | Net-Centric Computing |
| COMP2340 | Computer Systems Organization |
| COMP3161 | Database Management Systems |
| COMP3901 | Capstone Project |
| INFO2100 | Mathematics and Statistics for IT |
| INFO2110 | Data Structures for IT |
| INFO2180 | Web Design and Programming I |
| INFO3105 | Computer Systems and Administration |
| INFO3110 | Information Systems |
| INFO3155 | Information Assurance and Security |
| INFO3170 | User Interface Design for IT |
| INFO3180 | Dynamic Web Development II |
| **AND three (3) credits from Levels 2 or 3 courses offered by the Department of Computing, plus eighteen (18) credits from any discipline including Computing.** | |

|  |  |  |
| --- | --- | --- |
| SOFTWARE ENGINEERING[Mobile Application Technologies] (B.Sc.)  [Non-UGC Funded] \*Not being Offered 2022/2023\* | | |
| Introductory Courses  (Level 1) | **A B.Sc. in Software Engineering [Mobile Application Technologies] requires a total of forty-two (42) Level 1 credits from:** | |
| **Semester 1** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1220 | Computing and Society |
| COMP1210 | Mathematics for Computing |
| COMP1161 | Object-Oriented Programming |
| SWEN1003 | Current and Future Trends in Computing for Software Engineers |
| SWEN1005 | Mobile Web Programming |
| SWEN1006 | Research Methods for Software Engineers |
| SWEN1007 | Software Engineering Essentials |
| SWEN1008 | Technical Writing for Software Engineers |
| CHIN1001 | Chinese (Mandarin) 1A |
|  | CHIN1002 | Chinese (Mandarin) 1B |
|  | FOUN1001 | English for Academic Purposes |
|  | FOUN1101 | Caribbean Civilization |
| Advanced Courses  (Levels 2, 3 and 4) | **A B.Sc. in Software Engineering [Mobile Application Technologies] requires all of the following courses from Levels 2, 3 and 4:** | |
| CHIN2001 | Elementary Chinese Culture and Language |
| CHIN2002 | Intermediate Chinese Culture and Language |
| COMP2140 | Software Engineering |
| COMP2171 | Object Oriented Design and Implementation |
| COMP2190 | Net-Centric Computing |
| COMP2201 | Discrete Mathematics for Computer Science |
| COMP2211 | Analysis of Algorithms |
| COMP2340 | Computer Systems Organization |
| COMP3161 | Introduction to Database Management Systems |
| COMP3912 | Internship in Computing II |
| SWEN2165 | Requirements Engineering |
| SWEN3000 | Application Development for iOS Devices |
| SWEN3001 | Android Application Development I |
| SWEN3002 Android Application Development II | |
| SWEN3003 | Web & Mobile Application Development I |
| SWEN3004 Web & Mobile Application Development II | |
| SWEN3120 | Software Architecture |
| SWEN3130 | Software Project Management |
| SWEN3145 | Software Modeling |
| SWEN3165 | Software Testing |
| SWEN3185 | Formal Methods and Software Reliability |
| SWEN3920 Software Engineering Capstone Project | |
| SWEN4001 | Advanced Database Systems |
| SWEN4002 | IT Certification I |

|  |  |  |
| --- | --- | --- |
| COMPUTER SCIENCE (MAJOR) | | |
| Introductory Courses  (Level 1) | **A major in Software Computer Science requires a total of fifteen (15) Level 1 credits from:** | |
| COMP1210 | Mathematics for Computing |
| COMP1220 | Computing and Society |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| Advanced Courses  (Levels 2 and 3) | **A major in Computer Science requires a minimum of thirty-nine (39) credits from Computing courses at Levels 2 and 3 and must include:** | |
| COMP2140 | Software Engineering |
| COMP2171 | Object Oriented Design and Implementation |
| COMP2190 | Net-Centric Computing |
| COMP2201 | Discrete Mathematics for Computer Science |
| COMP2211 | Analysis of Algorithms |
| COMP2340 | Computer Systems Organization |
| COMP3101 | Operating Systems |
| COMP3161 | Introduction to Database Management Systems |
| COMP3220 | Principles of Artificial Intelligence |
| COMP3901 | Capstone Project |
|  | **AND nine (9) credits from Level 2 or 3 courses offered by the Department of Computing.** | |

|  |  |  |
| --- | --- | --- |
| SOFTWARE ENGINEERING (MAJOR) | | |
| Introductory Courses  (Level 1) | **A major in Software Engineering requires a total of fifteen (15) Level 1 credits from:** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| COMP1210 | Mathematics for Computing |
| COMP1220 | Computing and Society |
| Advanced Courses  (Levels 2 and 3) | **A major in Software Engineering requires a minimum of thirty-nine (39) credits from Levels 2 and 3 and must include:** | |
| COMP2140 | Software Engineering |
| COMP2171 | Object Oriented Design and Implementation |
| COMP2190 | Net-Centric Computing |
| COMP2201 | Discrete Mathematics for Computer Science |
| COMP2211 | Analysis of Algorithms |
| COMP3911 | Internship in Computing |
| SWEN3130 | Software Project Management |
| SWEN3145 | Software Modelling |
| SWEN3165 | Software Testing |
| SWEN3185 | Formal Methods and Software Reliability |
| SWEN3920 | Capstone Project  (Software Engineering) |
|  | **AND three (3) credits from Level 2 or 3 courses offered by the Department of Computing.** | |

|  |  |  |
| --- | --- | --- |
| COMPUTER SCIENCE (MINOR) | | |
| Introductory Courses  (Level 1) | **A minor in Computer Science requires a total of twelve (12) Level 1 credits from:** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| COMP1210 | Mathematics for Computing |
| Advanced Courses  (Levels 2 and 3) | **A minor in Computer Science requires a minimum of fifteen (15) credits from Levels 2 and 3 and must include:** | |
| COMP2201 | Discrete Mathematics for Computer Science |
| COMP2340 | Computer Systems Organization |
| **AND any three (3) courses from below:** | |
| COMP2010 | Probability and Statistics for Computing |
| COMP2120 | Digital Logic Design |
| COMP2130 | Systems Programming |
| COMP2140 | Software Engineering |
| COMP2171 | Object Oriented Design and Implementation |
| COMP2190 | Net-Centric Computing |
| COMP2211 | Analysis of Algorithms |
| COMP3101 | Operating Systems |
| COMP3220 | Principles of Artificial Intelligence |
| COMP3652 | Language Processors |
| COMP3702 | Theory of Computation |
| COMP3801 | Real-Time Embedded Systems |
| COMP3911 | Internship in Computing |

|  |  |  |
| --- | --- | --- |
| INFORMATION TECHNOLOGY (MINOR) | | |
| Introductory Courses  (Level 1) | **A minor in Information Technology requires a total of twelve (12) Level 1 credits from:** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| COMP1210 | Mathematics for Computing |
| Advanced Courses  (Levels 2 and 3) | **A minor in Information Technology requires a minimum of fifteen (15) credits from Levels 2 and 3 and must include:** | |
| COMP2190 | Net-Centric Computing |
| INFO2110 | Data Structures for IT |
| **AND any three courses from below:** | |
| INFO2100 | Mathematics and Statistics for IT |
| INFO2180 | Dynamic Web Development I |
| INFO3105 | Computer Systems and Administration |
| INFO3155 | Information Assurance and Security |
| INFO3170 | User Interface Design for IT |
| INFO3180 | Dynamic Web Development II |
| INFO3435 | eCommerce |

|  |  |  |
| --- | --- | --- |
| SOFTWARE ENGINEERING (MINOR) | | |
| Introductory Courses  (Level 1) | **A minor in Software Engineering requires a total of twelve (12) Level 1 credits:** | |
| COMP1126 | Introduction to Computing I |
| COMP1127 | Introduction to Computing II |
| COMP1161 | Object-Oriented Programming |
| COMP1210 | Mathematics for Computing |
| Advanced Courses  (Levels 2 and 3) | **A minor in Software Engineering requires a minimum of fifteen (15) credits from Level 2 and 3 and must include:** | |
| COMP2140 | Software Engineering |
| COMP2171 | Object Oriented Design and Implementation |
| **AND any three (3) courses from below:** | |
| COMP2201 | Discrete Mathematics for Computer Science |
| SWEN3130 | Software Project Management |
| SWEN3145 | Software Modelling |
| SWEN3165 | Software Testing |
| SWEN3185 | Formal Methods and Software Reliability |

##### COURSE DESCRIPTIONS

**COMP1126**

**INTRODUCTION TO COMPUTING I**

(3 Credits) (Level 1) (Semesters 1 or 2)

**Pre-requisites:**

A CAPE (Units 1 & 2 {or A-level}) Science subject,

ECON1003,

Teacher’s College Diploma,

Associate Degree in Mathematics or Science **OR** Information Technology.

**Course Content:**

1. **History of Programming Languages:** Brief survey of programming paradigms.
2. Building Abstractions.
3. **Computational Processes:** Primitive Operations, Special Forms for naming, conditional execution, Procedures as sequences of operations, Recursion and Iteration, Lexical scoping and Nested Procedures.
4. **Higher-order Procedures:** Customising Procedures with procedural arguments, Creating new functions at run-time.
5. **Compound Data**: Pairs and Lists.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
  + 1 Quiz 5%
  + 1 In-course Test (1 hour) 10%
  + 5 Laboratories 10%
  + 1 Written Assignment/ Programming Project 15%

**COMP1127**

**INTRODUCTION TO COMPUTING II**

(3 Credits) (Level 1) (Semesters 1 or 2)

**Pre-requisite:**

A CAPE (Units 1 & 2 {or A-level}) Science subject,

ECON1003,

Teacher’s College Diploma,

Associate Degree in Mathematics or Science **OR** Information Technology.

**Course Content:**

1. **Building Abstractions:** Compound Data (Lists and Trees); Abstract Data Types.
2. **Controlling Interactions:** Generic operations; Self-Describing Data; Message Passing; Streams and Infinite Data Structures; Object-oriented Programming.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
* 2 Quizzes 5%
* 1 In-course Test (1 Hour) 10%
* 5 Laboratories 10%
* 1 Written Assignment/ Programming Project 15%

**COMP1161**

**OBJECT-ORIENTED PROGRAMMING**

(3 Credits) (Level 1) (Semesters 1 or 2)

**Pre-requisites:**

COMP1126 - Introduction to Computing I **AND**

COMP1127 - Introduction to Computing II.

**Course Content:**

1. **Object-Oriented Programming:** Objects and Classes (Methods, Message Passing, Instance and Class Variables); Encapsulation and Information-Hiding; Imperative Control Structures, Assignment/State, Parameter Passing Models; Primitive Types, Inheritance, Polymorphism, Class Hierarchies; Object Composition; Abstract and Concrete Classes; Interfaces. Templates; Using APIS, Class Libraries, Modules/Packages; Array and String Processing; I/O Processing; Concept of Object References and Aliases; Collection Classes and Iterators; OO Testing. Debugging Tools.
2. **Graphics and GUI Programming, Web Concepts and Objects:** Introduction to GUI programming; Event-driven programming; Exception handling; Use of simple graphical libraries; and simple animation programming; Simple HTML-embedded objects such as applets.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* 3 Laboratories 5%
* 2 In-course Tests (1 hour each) 15% (5% & 10%)
* 3 Projects 30% (10% each)

**COMP1210**

**MATHEMATICS FOR COMPUTING**

(3 Credits) (Level 1) (Semesters 1 or 2)

**Pre-requisite:**

CSEC Mathematics.

**Course Content:**

Propositional Logic; Logical Connectives; Truth Tables; Normal Forms (Conjunctive And Disjunctive); Validity; Predicate Logic; Universal and Existential Quantification; Modus Ponens and Modus Tollens; Limitations of Predicate Logic; Functions (Surjections, Injections, Inverses, Composition); Relations (Reflexivity, Symmetry, Transitivity, Equivalence Relations); Sets (Venn Diagrams, Complements, Cartesian Products, Power Sets); Pigeonhole Principle; Cardinality and Countability; Finite Probability Space, Probability Measure, Events; Conditional Probability, Independence; Trees, Undirected Graphs, Directed Graphs, Spanning Trees/Forests.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
* 1 In-course Test 10%
* 3 Assignments/Quizzes 30% (10% each)

**COMP1220**

**COMPUTING AND SOCIETY**

(3 Credits) (Level 1) (Semesters 1 or 2)

**Pre-requisite:**

None.

**Course Content:**

1. **History of Computing:** History of computer hardware, software, networking; Regional computing history; Pioneers of computing. Contributions of region and of other developing countries.
2. **An Overview of Computing:** How hardware, software, and networks work at a conceptual level; use and high-level construction of computing artefacts, e.g., simple webpages, animations, robotics programs; Sub-disciplines within Computing: Computer Science, IT, IS, etc.; he global computing industry and its impact on industry and society; The use of computing in enterprise, entrepreneurship, various disciplines and careers.
3. **Social Context of Computing:** Social implications of computing and networked communication in general and on youth, e.g. cultural, self-image, possible effects of videogames; Understanding the social and cultural context of design; Understanding the potential of computing to transform society positively, globally or regionally, or to exacerbate inequalities or mask underdevelopment; Analysis of the government and business policies of developing and developed countries with successful computing industries; Accessibility issues in computing professions (e.g. class, culture, ethnicity, gender, disabled); Public policy issues (e.g. cyber-crime, privacy, electronic voting); Growth and control of and access to the Internet; Environmental Issues and Computing, e.g. e-waste, green computing.
4. **Professional Ethics in Computing:** Making and evaluating ethical choices and arguments, identifying assumptions and values; The nature of professionalism (including care, attention and discipline, fiduciary responsibility, and mentoring); Keeping up-to-date as a professional (in terms of knowledge, tools, skills, legal and professional framework as well as the ability to self-assess and computer fluency); Various forms of professional credentialing and the advantages and disadvantages; The role of the professional in public policy; Maintaining awareness of consequences of decisions; Introduction to ethics, ethical dissent and whistle-blowing; Codes of ethics, conduct, and practice (IEEE, ACM, SE, and so forth); Harassment and discrimination, “Acceptable use” policies for computing in the workplace; Healthy computing environment (ergonomics).
5. **Risks of Computing Products:** Historical examples of software risks (such as the Therac-25 case); Implications of software complexity on risk. The limits of computing.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* 2 Tutorial Presentations 20% (10% each)
* 3 Written Assignments 30% (10% each)

**SWEN1007**

**SOFTWARE ENGINEERING ESSENTIALS**

(3 Credits) (Level 1) (Semester 2)

**Pre-requisite:**

None

**Course Content:**

* Dynamics of working in teams and groups: Differentiate between team and group; team and group communication; reading, understanding, and summarizing reading; presentation skills (goals, slide composition, audience interaction); dealing with multicultural environments
* Individual cognition
* Accreditation, certification, and licensing: codes of ethics and professional conduct; the nature and role of software engineering standards; employment contracts
* Software engineering basics: life cycle, the four common activities; basic human considerations for code; software product basics
* Software engineering careers (including software entrepreneurs)
* Characteristics of successful/unsuccessful software engineering projects
* Engineering foundations: measurement and metrics; theory of measurement (e.g., criteria for valid measurement); engineering design (e.g., formulation of problem, alternative solutions, and feasibility)
* Software quality: software quality concepts and models; software quality assurance methods; software quality metrics; product quality attributes; software reliability; configuration control

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
* Project 40%
* Assignments 20%

**COMP2130**

**SYSTEMS PROGRAMMING**

(3 Credits) (Level 2) (Semester 1)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I **AND**

COMP1161- Object-Oriented Programming.

**Course Content:**

1. **Introduction to Computer Systems and UNIX Development Tools:** C Basics, UNIX development tool (gcc, gdb)**;** Using system libraries**;** Bits, bytes, and bitwise operators; Data structure and object implementation in C and C++; C pointers and arrays, C strings, malloc, realloc, and free as raw memory allocators Linked structures in C, C++; Data type and polymorphism, the void \*, function pointers, and generic functions; Floating point representation.
2. **Assembly Code:** Introduction to IA32, ALU operations, addressing, arithmetic, opcodes; Using gcc to generate your compilation product; Analysing compiled programs with gdb to understand the layout of data, functions, function calls, parameters, dynamic memory, etc.; Control function calls, runtime stack, passing by value and by address; C++ methods, the *this* pointer, references, RTTI, runtime and memory model for C++ objects and methods; Calling service routines.
3. **Memory Layout, Synthesis, and Execution of a UNIX Process:** Address spaces, implementations of malloc, realloc, and free; The compilation tool chain, linkers, loaders, and address space; Memory hierarchies, caches, locality, and pipelining; Programming for optimal use of caches and virtual memory; Writing simple optimised code, using gdb and profilers to analyse simple optimised compile programs; Heap allocation, implementation, and garbage collectors.
4. **Foreign Function Calls,** e.g., Java Native Interface (JNI).

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* 5 Assessed Tutorials 5%
* In-course Examination, (1 hour) 10%
* 10 Assessed Laboratories 10%
* 3 Programming Exercises 25%

**COMP2140**

**SOFTWARE ENGINEERING**

(3 Credits) (Level 2) (Semester 1)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I **AND**

COMP1161- Object-Oriented Programming.

**Course Content:**

1. **Software Design:** Fundamental design concepts and principles; The role and the use of contracts; Structured design; Design qualities; Internal - including low coupling, high cohesion, information hiding, efficiency; External - including reliability, maintainability, usability, performance.
2. **Using APIs:** Programming using APIs.
3. **Tools and Environments:** Programming environments; Requirements analysis and design modelling tools; Testing tools including static and dynamic analysis tools; Tools for source control, and their use in particular in team-work; Configuration management and version control tools; Tool integration mechanisms.
4. **Software Processes:** Software life-cycle and process models; Software process capability maturity models; Approaches to process improvement; Process assessment models; Software process measurements.
5. **Requirements Specifications:** Systems level considerations; Software requirements elicitation; Requirements analysis modelling techniques; Functional and non-functional requirements; Acceptability of certainty/uncertainty considerations regarding software/system behaviour; Prototyping.
6. **Software Verification Validation:** Distinguishing between verification and validation; Static approaches and dynamic approaches; Validation planning; documentation for validation; Different kinds of testing – human computer interface, usability, reliability, security, conformant to specification; Testing fundamentals, including test plan creation and test case generation black-box and white-box testing techniques; Defect seeding; Unit, integration, validation, and system testing; Measurements: process, design, program; Verification and validation of non-code (documentation, help files, training materials); Fault logging, fault tracking and technical support for such activities; Regression testing; Inspections, reviews, audits.
7. **Software Evolution:** Software maintenance; Characteristics of maintainable software; Reengineering Legacy systems; Refactoring.
8. **SE/Software Project Management:** Team management; Team processes; Team organization and decision-making; Roles and responsibilities in a software team; Role identification and assignment; Project tracking; Team problem resolution; Project scheduling; Software measurement and estimation techniques; Risk analysis ( The issue of security, High integrity systems, safety critical systems, The role of risk in the life cycle); Software quality assurance (The role of measurements); Software configuration management and version control; release management; Project management tools; Software process models and process measurements.
9. **Professional Ethics:** Community values and the laws by which we live; The nature of professionalism (including care, attention and discipline, fiduciary responsibility, and mentoring);Keeping up-to-date as a professional (in terms of knowledge, tools, skills, legal and professional framework as well as the ability to self-assess and computer fluency);Various forms of professional credentialing and the advantages and disadvantages**;** The role of the professional in public policy**;** Maintaining awareness of consequences**;** Ethical dissent and whistle-blowing**;** Codes of ethics, conduct, and practice (IEEE, ACM, SE, AITP, and so forth)**;** Dealing with harassment and discrimination**;** “Acceptable use” policies for computing in the workplace**;** Healthy computing environment (ergonomics).
10. **Risks:** Historical examples of software risks (such as the Therac-25 case); Implications of software complexity; Risk assessment and risk management; risk removal, risk reduction and risk control.

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%

One Software Development Group Project

* Requirements Documentation 15%
* Design Model (e.g., UML diagrams) 15%
* Presentations (10) using relevant tools 15%  
  e.g. PowerPoint
* Final Presentation of Implemented System 15%

**COMP2171**

**OBJECT ORIENTED DESIGN AND IMPLEMENTATION**

(3 Credits) (Level 2) (Semester 2)

**Pre-requisites:**

COMP1161 - Object-Oriented Programming **AND**

COMP2140 - Software Engineering.

**Course Content:**

1. **Fundamentals of Object Orientation**: Abstraction, Encapsulation, Information hiding, Coupling, Cohesion, Law of Demeter.
2. **Identifying Classes**: Domain Analysis, Systems Analysis, Class/Responsibility/Collaboration Cards (CRC Cards), Noun Verb Analysis.
3. **Identifying Class Relationships**: Dependencies, Associations, Aggregations, Compositions, Association Classes.
4. **Objects and relationships between objects**: Links and object diagrams.
5. **Modelling**: History of Modelling, Modelling Benefits, Agile Modelling, UML Diagrams: Use Case, Sequence, Communication, State, Activity, Class, Component, Deployment, Timing etc., Views: 4+1 views, Dynamic vs. Static etc. Design Patterns, Object Constraint Language.
6. **Tools:** e.g. Rational Software Architect, StarUML, Enterprise Architect, Visual Paradigm, Validating models, Other useful features of modelling tools.
7. **Software Architecture**: Definition, rationale, benefits, business and technical impact etc., Architectural patterns Emerging Topics in Object Oriented Design, Model Driven Engineering.

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
* Online Activities 10%
* In-course Test 15%
* Group Presentations 35%

**COMP2190**

**NET CENTRIC COMPUTING**

(3 Credits) (Level 2) (Semester 2)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I,

COMP1161 - Object-Oriented Programming **AND**

COMP1210 - Mathematics for Computing **or** MATH1152 - Introduction to Formal Mathematics).

*May not be credited with COMP3150 - Computing Networking and Communications.*

**Course Content:**

1. **Introduction:** Background and history of network and the Internet; Network architectures; Networks and protocols; Client/server and peer-to-peer paradigms; Mobile and wireless computing.
2. **Network Communication:** Network standards and standardization bodies; The ISO 7-layer reference model in general and its instantiation in TCP/IP; Overview of physical and data link layer concepts (framing, error control, flow control, and protocols); Data link layer access control concepts; Internetworking and routing (routing algorithms, internetworking, and congestion control); Transport layer services (connection establishment, performance issues, flow and error control); Web protocols with particular emphasis on HTTP.
3. **Distributed Computing.**
4. **Network Security:** Fundamentals of cryptography (Secret-key algorithms, Public-key algorithms); Authentication protocols, Network attack types, e.g., denial of service, flooding, sniffing, and traffic redirection; Basic network defence tools and strategies (Intrusion detection, Firewalls, Detection of malware, Kerberos, IPSec, Virtual Private Networks, Network Address Translation).
5. **Web Technologies:** Basic server-side programs (php, MySQL)**,** Basic client-side scripts (XHTML, XML, JavaScript, CSS)**,** Nature of the client-server relationship**,** Support tools for Web site creation and Web management.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + 7 Quizzes 5%
  + In-course Examination (1 hour) 10%
  + 2 Assignments 10%
  + 2 Projects 25%

**COMP2201**

**DISCRETE MATHEMATICS FOR COMPUTER SCIENCE**

(3 Credits) (Level 2) (Semester 1)

**Pre-requisite:**

COMP1210 - Mathematics for Computing **OR**

MATH1152 - Introduction to Formal Mathematics.

**Course Content:**

1. **Basics of Counting:** Arithmetic and geometric progressions; Fibonacci numbers; The pigeonhole principle; Basic definitions; Pascal’s identity; The binomial theorem; The Master theorem.
2. **Asymptotic Analysis:** Limits; Orders of Growth (Big-oh O, Omega Ω and Theta Θ).
3. **Graph Theory:** Trees; Planarity; Eulerian and Hamiltonian Cycles; Matching and Colouring.
4. **Elementary Probability Theory:** Counting in event space; Probability Tree; Probability distributions; Finite probability space, probability measure, events; Conditional probability, independence, Bayes’ theorem; Integer random variables, expectation; Law of large numbers.
5. **Generating Functions:** Convergence Properties; Convolution; Applications.
6. **Recurrence Relations.**
7. **Introduction to Automata, Grammars and Languages:** Finite-state machines; Context-free grammars**;** Language type classification and grammar type.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
  + 2 Quizzes 5%
  + In-course Test (1 hour) 15%
  + 4 Assessed Homework Assignments 20%

**COMP2211**

**ANALYSIS OF ALGORITHMS**

(3 Credits) (Level 2) (Semester 2)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I,

COMP1161 - Object-Oriented Programming **AND**

COMP1210 - Mathematics for Computing.

**Course Content:**

Analysing algorithms (solving recurrence equations with the Master Theorem); Algorithm strategies (brute force, greedy, divide, and conquer, branch-and bound, heuristic; Iterated approximations (Newton = Raphson method, searching for roots of a polynomial {in one variable}); Fast exponentiation; Euclid’s algorithm; Discrete logarithm; RSA cryptograph; Heaps as implementations for priority queues; Sorting; Binary search trees; Red-Black trees; Hashing; Graphs and graph algorithms; Distributed computing (introduction {consensus vs. election algorithms}); NP Basic Computability: uncomputable functions, the halting problem implicated of uncomputability.

**Evaluation:**

* Final Examination (2 hours)              50%
* Coursework:                                                   50%
  + 1 In-course Examination                    10%
  + 3 Written Homework Assignments   40%

**COMP2340**

**COMPUTER SYSTEMS ORGANIZATION**

(3 Credits) (Level 2) (Semester 2)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I,

COMP1161 - Object-Oriented Programming **AND**

COMP1210 - Mathematics for Computing.

**Course Content:**

1. **Data Representation and Digital Logic:** Overview of the history of the digital computer; Introduction to digital logic (logic gates, flip-flops, circuits); Representation of numeric data (floating point); Range, precision, and errors in floating-point arithmetic; Characters, pointers, strings, composite data (arrays, lists, objects).
2. **The Microarchitecture Level:** The functional units of the processor (adders, ALU’s, registers, buses); Data paths, microinstructions, the control unit; Hardwired controllers and micro-coded controllers.
3. **Instruction Set Architectures:** Introduction to instruction set architecture, microarchitecture and system architecture; Processor architecture (instruction types, register sets, addressing modes); Processor structures (memory-to-register and load/store architectures); Instruction sequencing, flow-of-control, subroutine call and return mechanisms; Structure of machine-level programs; Limitations of low-level architectures; Low-level architectural support for high-level languages; Translation (compiling, assembling, linking, loading).
4. **Peripherals and Protocols:** I/O fundamentals: handshaking and buffering; polling; Interrupt mechanisms: vectored and prioritized, interrupt acknowledgment; Buses: protocols, arbitration, direct-memory access (DMA), Examples of modern buses: e.g., PCIe, USB, Hypertransport.
5. **Memory**: Storage systems and their technology (semiconductor, magnetic, optical); Memory hierarchy, latency and throughput; Cache memories: operating principles, replacement policies, multilevel cache, cache coherency; Storage standards (CD-ROM, DVD); Sound and audio, image and graphics, animation and video; Multimedia standards (audio, music, graphics, image, telephony, video, TV); The significance of power dissipation and its effects on computing structures.
6. **Input/Output Devices:** Input devices: mice, keyboards (text and musical), scanners, touchscreen, voice; Video displays and printers; Input transducers (temperature, pressure, position, movement).
7. **Parallelism:** Processor and system performance measures and their limitations; Instruction pipelining and instruction-level parallelism (ILP); Superscalar architectures; vector processors; array processors; VLIW; Multicore and multithreaded processors; GPU’s and special-purpose graphics processors; Flynn’s taxonomy (Multiprocessor structures and architectures); Amdahl’s law.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + 5 Quizzes 5%
  + 1 In-course Test 10%
  + 6 Laboratories 15%
  + 2 Assignments 20%

**COMP2802**

**SPEECH PROCESSING**

(3 Credits) (Level 2) (Semester 2) *(\*Not being offered 2022/2023\*)*

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I **AND**

ELET2460 - Signals and Systems.

**Anti-requisites:**

ELET2210 - Speech Processing.

**Course Content:**

Speaking; Hearing; Sounds and symbols; Articulatory and acoustic phonetics; Phonology; Prosody; Speech spectra; Sampling; Fourier transform; Linear filters; Linear prediction; Cepstral analysis.

**Evaluation:**

* Final Examination (2 hours) 30%
* Coursework: 70%
  + 2 Programming projects 50% (25% each)
  + 2 In-course tests 20% (10% each)

**INFO2100**

**MATHEMATICS AND STATISTICS FOR IT**

(3 Credits) (Level 2) (Semester 2)

**Pre-requisite:**

COMP1210 - Mathematics for Computing.

**Course Content:**

Describe the difference between stochastic and deterministic analysis; Explain the purpose and nature of statistical sampling; Distinguish between the concepts of mean, median and mode, and discuss the drawbacks of each as a descriptive statistic; Calculate the mean, median and mode of a given sample of data; Calculate the standard deviation of a given sample of data; Explain, with examples, the role of probability and statistics in IT; Perform a statistical analysis of a system’s performance; Statistical analysis of a system’s performance and recommend ways to improve performance; Randomness, finite probability space, probability measure, events; Conditional probability, independence, Bayes’ theorem; Integer random variables, expectation; Formulation of hypotheses: null and alternate hypothesis; Parametric and non-parametric tests and their applicability; Criteria for acceptance of hypotheses, significance levels; t-test, z-test, Chi-square test, and their applicability; Correlation coefficients; Linear and nonlinear regression models; Stochastic versus deterministic analysis; Purpose and nature of sampling, its uses and applications; Mean, median, mode, variance, standard deviation.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
* 1 In-course Test (1 hour) 10%
* 3 Assignments/Quizzes 30% (10% each)

**INFO2110**

**DATA STRUCTURES FOR IT**

(3 Credits) (Level 2) (Semester 1)

**Pre-requisite:**

COMP1126 - Introduction to Computing I **AND**

COMP1127 - Introduction to Computing I **AND**

COMP1161- Object-Oriented Programming.

**Anti-requisite:**

COMP2211 - Analysis of Algorithms.

**Course Content:**

Primitive types; Arrays; Records; Strings and string Processing; Data representation in Memory; Pointers and References; Linked Structures; Knowledge of Hashing Function; Use of Stacks, Queues; Use of Graphs and Trees; Strategies for choosing the right Data Structure.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
* 1 In-course test (1 hour) 5%
* 3 Written assignments 15% (5% each)
* 2 Programming projects 20% (10 each)

**INFO2180**

**DYNAMIC WEB DEVELOPMENT I**

(3 Credits) (Level 2) (Semester 2)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I **AND**

COMP1161- Object-Oriented Programming.

**Course Content:**

Networking concepts, Internet protocols - TCP/IP. DNS, MIME types; XHTML, dynamic XHTML, CSS, DOM. XML, XSLT; Overview of website design principles (requirements, concept design, implementation, testing); Overview of website UI design: low-fidelity prototyping, layout, use of colour, fonts, controls; Server-side frameworks and languages, client-side languages. Basic session tracking; Introduction to three-tier architecture; Fundamental web frameworks and design patterns for the web; Overview of web server architecture and web services standards; Web database connectivity; Overview of principles, design and frameworks for e-commerce; Overview of network security issues, ethical and social issues; Introduction to multimedia for the web; Introduction to mobile and wireless web platforms.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* 1 In-course test (1 hour) 5%
* 10 Laboratories 10% (1% each)
* 5 Programming Projects 35% (7% each)

**SWEN2165**

**REQUIREMENTS ENGINEERING**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisite:**

COMP2140 - Software Engineering.

**Course Content:**

1. **Interacting with stakeholders**: dealing with uncertainty and ambiguity, negotiation, requirements attributes (complete, traceable, unambiguous, atomic), cognitive problem complexity elicitation tools and techniques under various development approaches (plan-driven, incremental, reuse, prototyping, and viewpoints).
2. **Requirements evolution:** prioritization, trade-off analysis, risk analysis, and impact analysis, evaluating cost-effective solutions, benefits realization, trade-off analysis, cost analysis, return on investment (ROI), change management, scope creep.
3. **Analyzing requirements:** safety, security, usability, performance**,** validating product quality, requirements interaction, functions, features, formal analysis.
4. **Requirements documentation:** types, audience, structure, quality, contemporary standards and best practices, software requirements specification techniques (decision tables, user stories, UML, Volere, behavioural specifications, goal-driven.
5. **Security in requirements analysis and specification.**
6. **Requirements engineering tools.**

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
  + One Group project 40%
  + Two Assignments (10% each) 20%

**COMP3029**

**AN INTRODUCTION TO QUANTUM COMPUTING**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

COMP2211 - Analysis of Algorithms **OR**

INFO2110 - Data Structures for IT **OR**

PHYS2351 - Quantum Mechanics and Nuclear Physics

**Course Content:**

1. **Foundations of quantum computing:** Why quantum computing - problems not in P; sub-atomic particles and their observables; photon, electron; polarization, spin; the influence of measurement, Heisenberg Uncertainty Principle; qubits, contemporary approaches and future considerations for their creation and manipulation; quantum dot, NV centre, Majorama fermions; logical layers of a quantum computer; quantum chip, quantum to classical interface, error detection, quantum compiler.
2. **The mathematics of quantum computing:** vectors, matrices, Dirac notation; vector addition and scalar multiplication, matrix multiplication, linear combination; inner product and tensor product; basis vectors and Bloch sphere representation; X, Y and Z basis states; Hilbert space; conjugate transpose, Hermitian matrix, Unitary matrix; global and relative phase.
3. **Quantum computing principles:** superposition; entanglement; interference.
4. **Quantum gates and circuits:** Pauli gates; Hadamard gate; Toffoli, CNOT, Rx, Ry, Rz, S, S;
5. **Circuits and code:** drawing quantum circuit diagrams as solutions to small, well-defined problems; use a computer-based tool to create, test and execute circuit-based solutions; use a high-level programming language to write code that can be executed on a quantum simulator or quantum computer.
6. **Quantum algorithms and networks:** BB84 quantum key distribution algorithm; Deutsch’s algorithm; Grover’s search algorithm; Building a quantum network.

**Evaluation:**

* Coursework: 100%
* 4 Assignments 40%
* 3 In-course tests 40%
* 2 Labs 20%

**COMP3101**

**OPERATING SYSTEMS**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

COMP2340 - Computer Systems Organization.

**Course Content:**

1. **Overview of Operating Systems:** Role and purpose of the operating system; History of operating system development; Functionality of a typical operating system; Mechanisms to support client-server models, hand-held devices; Design issues (efficiency, robustness, flexibility, portability, security, compatibility); Influences of security, networking, multimedia, windows.
2. **Operating System Principles:** Structuring methods (monolithic, layered, modular, micro-kernel models); Abstractions, processes, and resources; Concepts of application program interfaces (APIs); Application needs and evolution of hardware/software techniques; Device organization; Interrupts: methods and implementations; Concept of user/system state and protection, transition to kernel mode.
3. **OS/Concurrency:** States and state diagrams; Structures (ready list, process control blocks, and so forth); Dispatching and context switching; The role of interrupts; Concurrent execution (advantages and disadvantages); The “mutual exclusion” problem and some solutions; Deadlock: causes, conditions, prevention; Models and mechanisms (semaphores, monitors, condition variables, rendezvous); Producer-consumer problems and synchronization; Multiprocessor issues (spin-locks, re-entrancy).
4. **Scheduling and Dispatch:** Pre-emptive and non-preemptive scheduling; Schedulers and policies; Processes and threads; Deadlines and real-time issues.
5. **Memory Management:** Review of physical memory and memory management hardware; Paging and virtual memory; Multilevel paging; Working sets and thrashing; Caching.
6. **Security and Protection:** Overview of system security; Policy/mechanism separation; Security methods and devices; Protection, access control, and authentication.
7. **File Systems:** Files (data, metadata, operations, organization, buffering, sequential, non-sequential); Directories (contents and structure); File systems (partitioning, mount/unmount, virtual file systems); Standard implementation techniques; Memory-mapped files; Special-purpose file systems; Naming, searching, access, backups.
8. **Device Management:** Characteristics of serial and parallel devices; Abstracting device differences; Buffering strategies; Direct memory access; Recovery from failures.
9. **System Performance Evaluation:** Policies for caching, paging, scheduling, memory management, security, and so forth; Evaluation models: deterministic, analytic, simulation, or implementation-specific; How to collect evaluation data (profiling and tracing mechanisms).
10. **Scripting:** Scripting and the role of scripting languages; Basic system commands; Creating and executing scripts, parameter passing.
11. **Trends in Operating Systems:** Overview of contemporary operating systems, mobile operating systems, Future trends in operating systems.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* 2 Assignments (5% each) 10%
* 2 In-course tests (10% each) 20%
* 2 Projects (variable weighting) 20%

**COMP3161**

**DATABASE MANAGEMENT SYSTEMS**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisites:**

COMP1126 - Introduction to Computing I,

COMP1127 - Introduction to Computing I,

COMP1210 - Mathematics for Computing **AND**

COMP1161 - Object-Oriented Programming.

**Course Content:**

1. **Information Management Concepts:** Basic information storage and retrieval concepts; Information capture and representation.
2. **Database Systems:** Components of database systems; Database architecture and data independence; Use of a declarative query language (SQL).
3. **Data Modelling:** Relational data models; Object-oriented models; Semi-structured data models.
4. **Relational Databases:** Relational algebra; Relational database design; Functional dependency; Decomposition of a schema; Normal forms; Multi-valued dependency.
5. **Query Languages:** Overview of database languages; SQL (data definition, query formulation, update, constraints, and integrity); Select-project-join; Subqueries; Querying XML; Stored procedures.
6. **Views and Indexes:** Basic structure of an index; Creating indexes with SQL; Materialized Views.
7. **Transaction Processing:** Transactions; Failure and recovery; Concurrency control.
8. **Distributed Databases:** MapReduce processing model; NoSQL systems.
9. **Advanced Topics:** Security and user authorization**;** Recursion**;** On-line analytical processing (OLAP)**;** Query optimisation.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* 8 Quizzes (equally weighted) 5%
* 1 In-course Test (1 hour) 10%
* 4 Assignments (equally-weighted) 10%
* 1 Programming Project 10%
* 4 Assessed Laboratories (equally-weighted) 15%

**COMP3162**

**DATA SCIENCE PRINCIPLES**

(3 Credits) (Level 3) (Semester 2)

**Pre-Requisite:**

COMP2201- Discreet Mathematics for Computer Science **OR**

INFO2100 - Mathematics and Statistics for IT **AND**

COMP2211- Analysis of Algorithms **OR**

INFO2110 - Data Structures for IT

**Course Content:**

1. **Mathematical background** (sets, basic statistics: description, prediction, inference).
2. **Motivation and Introductory concepts:** What are data?
3. **Data Quality Criteria**: Validity (type, range, cross-field, other constraints), Accuracy, Completeness, Consistency, Uniformity.
4. **The Data Science Process. Applying the Data Science Process using a high-level programming language:** Data Wrangling: extractions, parsing, joining, standardizing, augmenting, cleansing, consolidating and filtering.
5. **Data Cleaning (ETL):** Data Auditing: Analysis (mean, standard deviation, range), Eliminating Duplicates, Translation and Normalization – Data Smoothing Techniques.
6. **Describing data:** Exploratory Data Analysis (EDA) + Data Visualization: Summaries, aggregation, smoothing, distributions, accessing data via different interfaces, Building structure from a variety of data forms to enable analysis.
7. **Modelling:** Linear and Stochastic (understand notions of uncertainty, simulations, random number generator, etc.).
8. **Simulation w/wo data:** probabilistic and/or resampling based Algorithms.
9. Data Science application areas and case studies.

**COMP3191**

**PRINCIPLES OF COMPUTER NETWORKING**

(3 Credits) (Level 3) (Semester 1) *(\*Not being offered 2021/2022\*)*

**Pre-requisite:**

COMP2190 - Net Centric Computing.

**Course Content:**

1. **Architectural Principles:** Layering; Encapsulation; Packet switching; Naming; End-to-end principle; Finite state machines.
2. **Application Layer:** HTTP (caching and HTTP future); FTP; SMTP and electronic mail; DNS (recursion); Peer to peer applications; Socket programming in TCP and UDP.
3. **Transport Layer:** Connectionless transport: UDP, Principles of reliable data transfer; Connection-oriented transport (TCP, TCP Tahoe, TCP Reno, and TCP New Reno, Congestion Control (RTT estimation and Self-clocking), Rationale for AIMD; Networks and protocols; Client/server and peer-to-peer paradigms; Mobile and wireless computing.
4. **Network Layer:** Names and addresses: ARP, IPv4, IPv6, and NAT; Routing and flooding, source routing, and spanning trees; Routing algorithms: Bellman-Ford, Dijkstra; Routing: Intra-AS routing (RIP and OSPF), Inter-AS routing (BGP), and multicast.
5. **Physical and Link Layers:** Shannon capacity and modulation; Bit errors; FEC and Reed-Solomon; MAC (ALOHA and Slotted ALOHA, CSMA/CD); Ethernet and Virtual LANs; Wireless (How it is different from wireline communication); Wireless principles (CSMA/CA and RTS/CTS; IEEE 802.11).
6. **Multimedia Networking:** Course Content-delivery networks; Queuing disciplines; Quality of service in computer networks.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* In-course Examination (1 hour) 10%
* 7 Quizzes (equally weighted) 5%
* 2 Individual written assignments 10%
* 2 Individual projects (10% +15%) 25%

**COMP3192**

**IMPLEMENTATION OF COMPUTER NETWORKS**

(3 Credits) (Level 3) (Semester 2) *(\*Not being offered 2021/2022\*)*

**Pre-requisite:**

COMP3191 - Principles of Computer Networking.

**Course Content:**

1. **Direct Link Networks:** Encoding; Framing; Error Detection; Reliable Transmission; SONET; FDDI; Network Adapters; Ethernet; 802.11 Wireless Networks.
2. **Packet and Cell Switching:** Concepts; ATM; Switching Hardware; Bridges & Extended LANs.
3. **Internetworking:** Internetworking Concepts; Global Internet; IPv6; Internet Multicast; Domain Name Services.
4. **End-to-End Protocols:** Concepts; UDP; TCP; APIs and Sockets; RPCs Performance.
5. **End-to-End Data:** Presentation Formatting; Data Compression; Security.
6. **Congestion Control:** Issues; Queuing Disciplines; TCP Congestion Control; Congestion Avoidance.
7. **High Speed Networking:** Performance Issues; Advanced Services; Experiences.
8. **Voice Over IP**: Overview; Peer to Peer calling; Call Managers; Call Signalling; PBX and Call Attendant Functionality.
9. **Routing Protocols:** IGPs and EGPs; Overview of RIP and OSPF; Introduction to BGP.

**Evaluation:**

* Final Examination (2 hours) ` 40%
* Coursework: 60%
* In-course Examination (1 hour) 10%
* 13 Quizzes (equally weighted) 15%
* 13 Laboratory Reports 20%
* Weekly Participation 15%

**COMP3220**

**PRINCIPLES OF ARTIFICIAL INTELLIGENCE**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisites:**

COMP2201- Discrete Mathematics for Computer Science **AND**

COMP2211 -Analysis of Algorithms.

**Course Content:**

1. **Introduction to AI:** Overview and History of AI and Philosophical Issues in AI.
2. **Intelligent Agents:** Performance measures, Environment, Actuators and Sensors (PEAS); Environment types; Agent types.
3. **Search:** Uninformed Search Algorithms; Heuristic Search Algorithms; Iterative Improvement Algorithms; Game Playing.
4. **Knowledge Representation and Reasoning:** Logic; Production Rules; Differencing Mechanisms; Expert Systems.
5. **Current topics in AI:** Machine Learning; Neural Networks; Reasoning Under Uncertainty; Natural Language Processing; Speech Recognition; Robotics; Fuzzy Logic; Virtual Reality.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
* 1 In-course Test 10%
* 1 written Assignment 10%
* 1 Programming Assignment 10%
* 1 Research Paper 10%

**COMP3410**

**INTRODUCTION TO PARALLEL COMPUTING**

(3 Credits) (Level 3) (Semester 2) *(\*Not being offered 2021/2022\*)*

**Pre-requisites:**

COMP2201 - Discrete Mathematics for Computer Science **OR**

COMP2211 - Analysis of Algorithms **AND**

COMP2340 - Computer Systems Organization.

**Course Content:**

1. **Basic Techniques (Parallel Computers):** The demand for computational speed, Potential for increased computational speed, Types of parallel computers, Cluster computing.
2. **Parallel Hardware & Parallel Software:** Von Neumann architecture, Processors, multitasking, and threads, Parallel hardware, Parallel software, Performance, Parallel program design, Writing and running parallel programs.
3. **Message-Passing Computing:** Basic message-passing programming, Using a cluster of computers, Evaluating parallel programs.
4. **Partitioning & Divide-and-Conquer Strategies:** Partitioning, Partitioning& Divide-and-conquer examples, Distributed-Memory Programming with Parallel Virtual Machine, Compilation and execution, PVM programs, SPMD programs, Communication, Performance Evaluation of PVM programs, Synchronous Computations, Synchronization, Barrier, Tree implementation, Butterfly barrier, Local synchronization, Deadlock.
5. **Sorting Algorithms:** Compare-and-Exchange sorting, algorithms, Bubble sort, Merge (bitonic) sort, Merge sort.
6. **Numerical Algorithms:** Matrices, Matrix addition, Matrix multiplication, Matrix-Vector multiplication, Implementing matrix multiplication, Solving a system of linear equations, Iterative methods.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
* Group Programming Project 15%
* Two Assignments 15%
* Two Quizzes 20%

**COMP3606**

**WIRELESS & MOBILE COMPUTING**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisites:**

COMP2602 - Enterprise Database Systems (St. Augustine Campus)

**Course Content**:

1. **Wireless and Mobile Networks Theory:** Why study wireless networks? History and evolution of wireless standards; Special problems of wireless and mobile computing; Wireless LANs; Bluetooth; Cellular Networks; Mobile Internet Protocol; Software support for mobile and wireless computing (for mobile phones and/or other mobile devices)
2. **Current Wireless Programming Languages**
3. **Android:** Introduction to Android; Android Features; Android Operating Systems; Simple Android application and architecture; Setting up the environment.
4. **Creating Applications and Activities:** Architecture of an application; Activities and Life cycles; Running an application; Activity Practice.
5. **Building User Interfaces:** Widget introduction; Utilizing popular widgets to develop simple to intermediate level applications.
6. **Intents and Broadcast Receivers:** Introduction to Intents; Explicit Intents; Implicit Intents; Programming practice with Explicit Intents; Programming practice with Implicit Intents; Programming practice with combination of Explicit and Implicit Intents.
7. **Files, Saving State and Preferences:** Files in Internal and External Storage; Reading, writing and opening files from internal storage; File applications.
8. **Saving state and preferences within applications.**
9. **Hardware Sensors:** Use of some popular sensors (subject to availability on phones).
10. **Maps and Location-Based services**
11. **SMS:** Introduction; Client/Server SMS architecture; Client applications; Server applications; Combined client/server applications
12. **Emerging Technologies**

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + 2 Assignments 20%
  + In-class test 15%
  + In-class assignment 15%

**COMP3652**

**LANGUAGE PROCESSORS**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

COMP2211 - Analysis of Algorithms.

**Course Content:**

1. **Syntactic Processing:** Context Free Grammars: Definition, BNF notation, ambiguity, parse trees and derivations; Regular Expressions: Definition, JLex or JFlex (a lexing tool); Parsing (top down (recursive descent and LL (K)); Parsing (bottom up (LR (0), SLR, LALR (1) and LR (1) parsers).
2. **Semantic Representation and Processing:** Operational vs. Denotational semantics, POSTFIX: an example of a stack-based programming language, Syntax-directed interpretation (and translation), Abstract Syntax Trees as Intermediate Representations, Interpretation and translation by AST traversal.
3. **Features of Programming Languages:** Typing (static vs. dynamic); Scoping (static vs. dynamic); Evaluation (lazy vs. eager); Parameter passing conventions; Data allocation strategies; First class citizens (objects); Tail recursion; Garbage collection.

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
* Written Homework Assignment 10%
* Programming Assignment 20%
* Project 30%

**COMP3702**

**THEORY OF COMPUTATION**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisite:**

COMP2201- Discrete Mathematics for Computer Science.

**Course Content:**

1. **Computability:** Regular Languages (DFA, NFA, Regular Expressions)**;** Context Free languages (CFGs, PDAs); Turing-recognisable Languages (Turing Machines) Church-Turing thesis (Lambda Calculus); Turing Reducibility and Mapping Reducibility; Undecidability.
2. **Complexity Theory:** Distinction between Time and Space complexity; Definitions of Complexity Classes: L, P, NP, PSPACE, EXPTIME; Eﬀect of Nondeterminism on Space and Time Complexity; Polynomial Time Mapping Reducibility; Hardness and Completeness Relative to Various Complexity Classes (e.g. NP-hardness, NP-completeness); Example NP-complete problems.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + 1 In-course Test 10%
  + 5 Written Homework Assignment 40%

**COMP3801**

**REAL TIME EMBEDDED SYSTEMS**

(3 Credits) (Level 3) (Semester 1) (*\*Not being offered 2021/2022\*)*

**Pre-requisites:**

COMP2140 - Software Engineering **AND**

COMP2340 - Computer Systems Organisation.

**Course Content:**

1. **Sensors, Actuators and Electrical Components:** Analogue to Digital Conversion, Sensor Formatting, Sensor Input Modules; Actuator Selection, Embedded hardware components; Hardware components for signal processing.
2. **State, Control and Feedback:** State diagrams and Petri Nets; Control and Feedback; Controllers.
3. **Embedded Design:** Hardware/Software Co-design; Fault Tolerance.
4. **Real Time Operating Systems:** Real Time Operating Systems; RTOS Example, e.g., VxWorks.
5. **Robotics and Multi-platform Programming:** Introduction to Robotics; Introduction to Mobile Programming with J2ME; Developing and deploying mobile applications; Load Balancing in Embedded Systems.

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
  + 1 In-course Test 10%
  + 2 Written Assignments 10%
  + 4 Group Projects 40%

**COMP3802**

**SPEECH AND LANGUAGE TECHNOLOGY**

(3 Credits) (Level 2) (Semester 1) *(\*Not being offered 2021/2022\*)*

**Pre-requisites:**

COMP2802 - Speech Processing **OR**

ELET2210 - Speech Processing.

**Anti-requisite:**

ELET3211 - Speech and Language Technology.

**Course Content:**

1. Introduction to speech technology
2. Speech signal processing
3. Probability theory for speech processing
4. Hidden Markov models and deep neural networks for speech processing
5. Acoustic modelling
6. Language modelling
7. Approaches to decoding
8. Model adaptation
9. Speech recognition examples
10. Speaker identification technologies
11. Speech synthesis

**Evaluation:**

* Final Examination (2 hours) 30%
* Coursework: 70%
  + 2 Programming projects (2x25%) 50%
  + 2 In-course tests (2x10%) 20%

**COMP3901**

**CAPSTONE PROJECT**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisites:**

COMP2140 - Software Engineering,

COMP2211 - Analysis of Algorithms,

**AND** any 6 credits of Level 2 or 3 Computing code courses.

**Course Content:**

The specific technical topics covered by each group will depend on the type of project. Common examples of such topics include (but are not limited to) Database Design, Web Programming, User-Interface Design, Mobile Application Development, Algorithm Design.

**Evaluation:**

This course is assessed via a series of presentations and a demonstration, a written report and a webpage. The specific contribution of each component towards the overall grade for a group is as follows:

Coursework: 100%

* + Mid-semester Presentation 10%
  + Web Page 10%
  + Final presentation 15%
  + Final demonstration 15%
  + Final Report 50%

*The presentations, demonstrations and Web pages are assessed by the evaluation committee. Each group final report is assessed by its supervisor and group members peer-assess each other. This combined level of assessment allows for individual grading.*

**COMP3911**

**INTERNSHIP IN COMPUTING I**

(3 Credits) (Level 3) (Semester 1, 2 and Summer Term)

**Pre-requisite:**

Permission of the Head of Department.

**Course Content:**

The exact nature of the internship depends upon the interests of the student and the specific needs of the cooperating organisation. It is assumed and expected that the intern will be involved in some area of computing and thereby gain valuable experience in his/her selected field of study.

Internships contribute to the education of the whole person by emphasizing the importance of work and by providing opportunities for self-reflection. The internship should be chosen to build on the student's own interests and to relate what he/she has learned in school to its application in the workplace. In addition, the internship should help the student evaluate him/herself as a worker and as a potential employee in a particular professional field. Through the internship, the student will enhance his/her feelings of self-worth and confidence in performing in the workplace. While on the job, the student should not only apply lessons learned in school to his/her particular job tasks, but he/she should also explore vocational possibilities and seek to discover what kinds of work he/she enjoys. In addition, the student will be able to build on his/her résumé and professional portfolio. Internship experiences should also offer the student access to potential mentors in his/her professional field.

**Responsibility of the Student:**

The student is required to spend about l50 working hours (e.g. 12 hours per week for approximately 13 weeks during semester 1 or 2, or 40 hours per week for approximately 4 weeks) working on a project or projects of the participating organisation’s choice. Where the students are registered for the course in semester 1 or 2, the hours allotted for the internship exercise should be selected by the student, at times when no classes are scheduled.

**The student must:**

* meet regularly with the Departmental Internship Coordinator (IC) and periodically with fellow interns to discuss his/her internship experiences
* maintain a journal indicating dates and hours worked, and a brief description of the work performed
* submit a final report summarising and evaluating the internship experience; and
* complete a résumé and interview at the Office of Placement and Career Services, UWI (Mona)

Any problems encountered during the internship should be discussed immediately with the IC so that appropriate action can be taken.

**Responsibility of the participating Organisation:**

Participating organisations will be vetted by the Internship Coordinator to ensure that they are suitable.

**The organisation will:**

* provide a mentor and appropriate work environment
* expose the student to the type of work which he/she would encounter in an entry level professional position
* provide appropriate personnel to oversee the project(s) assigned to the student, and the resources needed to accomplish the work
* treat the student as it would any employee, and
* expect the same degree of responsibility from the student, even as the student is not an employee of the firm

**The mentor will be asked to:**

* provide a written evaluation of the student’s performance to the IC at the end of the internship;
* provide the student with a periodic evaluation of his/her performance; and
* consult with the IC when and if necessary.

Although an internship is a learning experience, it is expected that the student will normally earn some compensation for work performed that may contribute to income generating activities, either in the form of a wage, stipend, or reimbursement of expenses.

**Responsibility of the Internship Coordinator (IC):**   
The IC will:

* organise preparation seminars for students at the start of each semester., featuring presentations from the Office of Placement and Career Services, industry personnel and alumni
* arrange preliminary meetings with mentors where students are briefed on expectations and responsibilities specific to the organisation;
* meet/correspond with students: student group meetings (weekly) via online journal, videoconference, etc. for students to share experiences;
* review reports from the organisation;
* review reports from the student;
* serve as a liaison between the Department of Computing (DoC) and the participating organisation;
* oversee the progress of the intern;
* make suggestions to both the student and the organisation on ways to enhance the benefits of the internship;
* meet regularly with the intern to discuss his/her experiences
* help resolve any problems the organisation and the student might have; and
* review all the reports submitted by the participating organisation and the student.

**Evaluation:**

There will be two components of the course’s assessment: the internship mentor’s evaluation and the student’s work during the internship and his/her final submission at the conclusion of the internship. Students must pass both aspects of the course.

The internship mentor will provide a written evaluation of the student's performance. This assessment will be done using a 5-point Likert scale**.** An assessment/evaluation form will be provided for this purpose, and the form will be returned to the DoC in a sealed envelope. The internship coordinator will assign a grade not exceeding 25% of the possible marks based on this assessment, and on the student’s journal which would detail the tasks assigned to the student and their level of completion.

**The student will be evaluated on:**

* Quality of work;
* Use of time (efficient/effective use of time to complete tasks);
* Ability to take initiative (ability to work independently);
* Grasp of subject (understanding of applicable standards and procedures);
* Judgement skills (ability to make appropriate work-related decisions);
* Interpersonal relations/teamwork (effectiveness in working with peers and supervisors);
* Adaptability (ability to alter activities to accommodate change);
* Problem solving/critical thinking skills;
* Punctuality, attendance;
* Verbal and written communication skills;
* Whether the goals of the internship were met (qualitative response);
* What skills the student developed (qualitative response);
* The observed primary strengths of the intern (qualitative response);
* Recommendations for improvement (qualitative response);
* What is your overall assessment of the student’s performance? (qualitative response); and
* Other relevant observations.

**75% will be based on the following:**

* Regular communication with the DIC (weekly reports) - 15%
* Attendance at and participation in required internship meetings (weekly) - 10%;
* Oral presentation summarizing the activities completed during the internship - 20%
* Documentation of the internship experience in an internship portfolio (30%) which includes:
* A final report summarizing the internship, relating it to courses done, and reflecting on the experience. The final report will have an appendix containing the student’s journal entries from the internship (guidelines will be provided).
* An updated résumé that incorporates the internship experience.
* A "company evaluation form” rating the participating organisation.
* Proof of consultation/debriefing with the Office of Placement and Career Services, UWI (Mona).

**COMP3912**

**INTERNSHIP IN COMPUTING II**

(6 Credits) (Level 3) (Semester 1, 2 and Summer Term)

**Pre-requisite:**

Permission of the Head of Department.

**Course Content:**

The exact nature of the internship depends upon the interests of the student and the specific needs of the cooperating organisation. It is assumed and expected that the intern will be involved in some area of computing and thereby gain valuable experience in his/her selected field of study.

Internships contribute to the education of the whole person by emphasizing the importance of work and by providing opportunities for self-reflection. The internship should be chosen to build on the student's own interests and to relate what he/she has learned in school to its application in the workplace. In addition, the internship should help the student evaluate him/herself as a worker and as a potential employee in a particular professional field. Through the internship, the student will enhance his/her feelings of self-worth and confidence in performing in the workplace.

While on the job, the student should not only apply lessons learned in school to his/her particular job tasks, but he/she should also explore vocational possibilities and seek to discover what kinds of work he/she enjoys. In addition, the student will be able to build on his/her résumé and professional portfolio. Internship experiences should also offer the student access to potential mentors in his/her professional field.

**Responsibility of the Student:**

The student is required to spend about l50 working hours (e.g. 12 hours per week for approximately 13 weeks during semester 1 or 2, or 40 hours per week for approximately 4 weeks) working on a project or projects of the participating organisation’s choice. Where the students are registered for the course in semester 1 or 2, the hours allotted for the internship exercise should be selected by the student, at times when no classes are scheduled.

**The student must:**

* meet regularly with the Departmental Internship Coordinator (IC) and periodically with fellow interns to discuss his/her internship experiences
* maintain a journal indicating dates and hours worked, and a brief description of the work performed
* submit a final report summarising and evaluating the internship experience; and
* complete a résumé and interview at the Office of Placement and Career Services, UWI (Mona)

Any problems encountered during the internship should be discussed immediately with the IC so that appropriate action can be taken.

**Responsibility of the participating Organisation:**

Participating organisations will be vetted by the Internship Coordinator to ensure that they are suitable.

The organisation will:

* provide a mentor and appropriate work environment
* expose the student to the type of work which he/she would encounter in an entry level professional position
* provide appropriate personnel to oversee the project(s) assigned to the student, and the resources needed to accomplish the work
* treat the student as it would any employee, and
* expect the same degree of responsibility from the student, even as the student is not an employee of the firm

The mentor will be asked to:

* provide a written evaluation of the student’s performance to the IC at the end of the internship
* provide the student with a periodic evaluation of his/her performance; and
* consult with the IC when and if necessary.

Although an internship is a learning experience, it is expected that the student will normally earn some compensation for work performed that may contribute to income generating activities, either in the form of a wage, stipend, or reimbursement of expenses.

**Responsibility of the Internship Coordinator (IC):**   
The IC will:

* organise preparation seminars for students at the start of each semester., featuring presentations from the Office of Placement and Career Services, industry personnel and alumni;
* arrange preliminary meetings with mentors where students are briefed on expectations and responsibilities specific to the organisation;
* meet/correspond with students: student group meetings (weekly) via online journal, videoconference, etc. for students to share experiences;
* review reports from the organisation;
* review reports from the student;
* serve as a liaison between the Department of Computing (DoC) and the participating organisation;
* oversee the progress of the intern ;
* make suggestions to both the student and the organisation on ways to enhance the benefits of the internship;
* meet regularly with the intern to discuss his/her experiences;
* help resolve any problems the organisation and the student might have; and
* review all the reports submitted by the participating organisation and the student.

**Evaluation:**

There will be two components of the course’s assessment: the internship mentor’s evaluation and the student’s work during the internship and his/her final submission at the conclusion of the internship. Students must pass both aspects of the course.

The internship mentor will provide a written evaluation of the student's performance. This assessment will be done using a 5-point Likert scale**.** An assessment/evaluation form will be provided for this purpose, and the form will be returned to the DoC in a sealed envelope. The internship coordinator will assign a grade not exceeding 25% of the possible marks based on this assessment, and on the student’s journal which would detail the tasks assigned to the student and their level of completion.

**The student will be evaluated on:**

* Quality of work;
* Use of time (efficient/effective use of time to complete tasks);
* Ability to take initiative (ability to work independently);
* Grasp of subject (understanding of applicable standards and procedures);
* Judgement skills (ability to make appropriate work-related decisions);
* Interpersonal relations/teamwork (effectiveness in working with peers and supervisors);
* Adaptability (ability to alter activities to accommodate change);
* Problem solving/critical thinking skills;
* Punctuality, attendance;
* Verbal and written communication skills;
* Whether the goals of the internship were met (qualitative response);
* What skills the student developed (qualitative response);
* The observed primary strengths of the intern (qualitative response);
* Recommendations for improvement (qualitative response);
* What is your overall assessment of the student’s performance? (qualitative response); and
* Other relevant observations.

**75% will be based on the following:**

* regular communication with the DIC (weekly reports) - 15%
* attendance at and participation in required internship meetings (weekly) - 10%;
* oral presentation summarizing the activities completed during the internship - 20%;
* documentation of the internship experience in an Internship Portfolio (30%) which includes:
* A final report summarizing the internship, relating it to courses done, and reflecting on the experience. The final report will have an appendix containing the student’s journal entries from the internship (guidelines will be provided).
* An updated résumé that incorporates the internship experience.
* A "company evaluation form” rating the participating organisation.
* Proof of consultation/debriefing with the Office of Placement and Career Services, UWI (Mona).

**INFO3105**

**COMPUTER SYSTEM ADMINISTRATION**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

COMP2340 - Computer Systems Organization **AND**

COMP2190 - Net-Centric Computing.

**Course Content:**

1. **Operating Systems:** Overview; Operating system principles; Concurrency, Scheduling and dispatch; Memory Management; Device Management; Security and Protection; File Systems; Real-time and embedded systems; Fault tolerance; Scripting; Virtualisation; Installation, Configuration and Maintenance of OS and Applications; Installation and Configuration; Maintenance (upgrades, patches, etc.); Server services (print, file, DHCP, DNS, FTP, HTTP, mail, SNMP, telnet); Application Management (database, web, network services, etc.); Deployment of a System Image using Imaging Software; Support and Licensing issues.
2. **Administration Activities:** Content Management; Content Deployment (file system planning and Structure); Server Administration and Management; User and Group Management; Backup Management; Security Management; Disaster Recovery; Resource Management; Automation Management (automatic job scheduling); Use of Site Management Logs; System Support.
3. **Administrative Domains:** Web; Network; OS; Support; Database.
4. **Power Management:** Power Requirements for Individual Systems; Heat and Power Budgets; Power Load Monitoring and Management.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + 1 Programming Project 10%
  + 5 Laboratories (4% each) 20%
  + 2 Written Assignments (10% each) 20%

**INFO3110**

**INFORMATION SYSTEMS IN ORGANISATIONS**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisites:**

COMP2140 - Software Engineering **AND**

COMP2190 - Net-Centric Computing.

**Course Content:**

1. **Characteristics of an Organization:** Business Functions; Management Hierarchy; Business Processes.
2. **Information Systems:** Types of Applications; Enterprise Systems; Supply Chain Management Systems; Customer Relationship Management Systems; Knowledge Management Systems.
3. **Information Systems and Business Strategy:** Corporate Strategy; Information Systems Strategy; Strategic Information Systems.
4. **Information Technology Infrastructure:** Computer Hardware; System Software; Data Management; Telecommunication Networks.
5. **IT for Business Intelligence Gathering:** Data mining; Artificial Intelligence Environment Scanning.
6. **Internet and other IT Innovations:** E-Commerce; E-Business; Collaborative Commerce.
7. **Managing Information Systems:** Information Systems Security and Control; Disaster Planning and Recovery.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
  + In-course Test (4% each) 10%
  + 3 Written Assignments (10% each) 30%

**INFO3155**

**INFORMATION ASSURANCE & SECURITY**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisites:**

COMP2190 - Net-Centric Computing **AND**

INFO2100 - Mathematics and Statistics for IT **or**

COMP2201 - Discrete Mathematics for Computer Science.

**Course Content:**

The reality for the growing need of security in our day to day tasks; Confidentiality, Integrity and Availability (the pillars of security); The ethical issues facing the Security Professional; Physical access to Information Resources (secure sites, security policies, backups, disaster recovery); The Human Factor (social engineering); Malware (viruses, worms, Trojan horses, mailers etc.); Penetration testing (threat discovery, assessment and system hardening); Confidentiality, integrity and non-repudiation (the use of cryptography in security (hash functions, message digests, public/private key cryptography)).

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework 60%

**INFO3170**

**USER INTERFACE DESIGN FOR IT**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisites:**

COMP2140 - Software Engineer **OR** I

NFO2180 - Dynamic Web Development I.

**Course Content:**

1. **Overview of HCI:** The Role of User Interfaces in Computer Applications; History of Human-Computer Interaction (HCI) and User Interface (UI) Systems; Human Factors (Perception, Movement, and Cognition); Ergonomics; Contextual Issues in HCI (Culture, Communication, and Organizations); HCI Models, UI Paradigms (Command, Graphical User Interface (GUI) etc., UI Guidelines).
2. **UI Environments:** Overview of graphics systems, display devices, input devices; GUI system architecture, event-driven interaction model; UI toolkits; Collaborative Systems. Embedded Systems.
3. **UI Development Methods:** UI development cycle (investigation, design, prototyping, evaluation, implementation); Developing UI requirements (inquiry methods, developing task and workflow models); Information collection and analysis methods; Prototyping (storyboarding, implementation); Evaluation methods (heuristic, observational, empirical).

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + In-course Test 5%
  + Programming Projects 45%

**INFO3180**

**DYNAMIC WEB DEVELOPMENT II**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

INFO2180 - Dynamic Web Development.

**Course Content:**

DOM. XML, XSLT, AJAX; Web Application Design Principles (requirements, concept design, implementation, testing); Web Application UI design (low-fidelity prototyping, layout, use of colour, fonts, controls); Further Server-Side Frameworks and Languages, Client-Side Languages; Session Tracking; *n*-tier Architecture for the Web; Service-oriented Architectures; Web Frameworks and Design Patterns for the Web; Web Server Architecture and Web Services Standards; Principles, Design and Frameworks for E-Commerce; Web Security Issues (Cross-site Scripting, SQL Injection, Phishing); Web Network Security Issues, Ethical and Social Issues; Multimedia for the Web; Mobile and Wireless Web Platforms.

**Evaluation:**

* Final Examination (2 hours) 50%
* Coursework: 50%
  + In-course Test 5%
  + 10 Laboratories 10%
  + 5 Programming Projects (7% each) 35%

**INFO3435**

**E-COMMERCE**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisites:**

COMP2140 - Software Engineering **AND**

INFO2180 - Dynamic Web Development

**Course Content**:

eCommerce Business Models and Concepts; The Internet and World Wide Web; eCommerce Infrastructure; Building eCommerce Web Site; eCommerce Website Evaluation and Usability Testing (Personalization &Customization); Online Security and Payment Systems; Ecommerce Marketing Concepts Ecommerce Marketing Communications; Ethical, Social, and Political Issues in Ecommerce; Online Retailing and Services; Online Content and Media; Social Networks, Auctions, and Portals; B2B Ecommerce (Supply Chain Management and Collaborative Commerce).

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
  + In-course Test 10%
  + 3 Assignments 30%

**INFO3606**

**CLOUD COMPUTING**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisites:**

COMP2605 - Enterprise Database Systems (St. Augustine Campus)

**Course Content**:

1. Introduction to Virtualization (i. Review of various virtualization technologies, hosts, hypervisors; ii. Configuring servers with virtualization)
2. Managing Virtual Machine Hosts (i. Managing using interfaces; ii. Remote management)
3. Networking in virtualization systems (i. Network planning; ii. Virtual switches)
4. Load balancing and high availability (i. Setting up load balancing; ii. Setting up high availability systems; iii. Virtual machine migration)
5. Cloud Technologies Introduction (i. Cloud as a service; ii. Infrastructure as a Service (IaaS); iii. Platform as a Service (PaaS); iv. Software as a Service (SaaS); v. Other “as a Service” offering; vi. Business Processes as a Service (BPaaS); vii. Management as a Service (MaaS); viii. Business as a Service (BaaS))
6. On premise cloud systems (i. Creating an on-premise cloud system; ii. Managing an on-premise cloud system; iii. Running applications on cloud systems; iv. Automating cloud management)
7. Public cloud systems (i. Interaction with public cloud systems; ii. Running of applications on public cloud systems; iii. Interaction with public cloud systems with scripting and automation)
8. Future of Cloud (i. Big Data; ii. Storefronts as a Service)

**Evaluation:**

* Final Examination (1.5 hours) 20%
* Coursework: 80%
  + 2 Assignments 20%
  + 2 Practical Assessments 30%
  + Project Report 20%
  + Presentation 10%

**SWEN3000**

**APPLICATION DEVELOPMENT FOR iOS DEVICES**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisite:**

COMP2171 - Object Oriented Design and Implementation

**Course Content:**

1. **Introduction to development on MacOS’s Xcode IDE**
2. **Introduction to Swift:** Types, literals and subscripting Initializers, properties, instance methods; Optionals; Loops; String interpolation; Enumerations and raw values; Classes and methods; Inheritance, polymorphism, dynamic typing, dynamic binding; arrays, set, dictionaries; categories and protocols.
3. **Xcode and Interface builder:** Application lifecycle; Xib, Storyboard, and interface builder; creating and building simple applications; UIState preservation; view application sandbox and crash logs.
4. **Cocoa Design Patterns:** Model, View and Controller (MVC) classes; Delegate and data source; Singleton pattern; Observer pattern; Target-action; Cocoa coding standards.
5. **Views and the view hierarchy:** the view hierarchy; creating a new project; views and frames; Labels; The Auto Layout System; Constraints in Interface Builder; Intrinsic content size; Misplaced views.
6. **Memory Management:** alloc, init, retain, release; Auto-release pool**.**
7. **Text input and delegation:** Text editing; keyboard attributes; responding to text field changes; dismissing the keyboard, number formatters; delegation; conforming to a protocol; using a delegate.
8. **View controllers:** View of view controller; Setting the initial view controller; UITabBarController; Tab bar items; Loaded and appearing views; Accessing subviews; Interacting with view controllers and their views.
9. **Interaction with UIControls:** Button, label, text fields; Switch, slider, progress bar; Alerts, action sheet; Tableviews; Scrollview, Web view; Maps; Searchbar, Popovers; Picker, Date picker, ImageView, ImagePicker Controller; Gestures.
10. **UITableView and UITableViewController**: UITableViewController; subclassing UITableViewController; Item classes; Custom initializers; UITableView’s Data Source; Implementing data source methods; Creating and retrieving UITableViewCells; Reusing UITableViewCells; Content insets; Editing UITableView; User Alerts
11. **Orientation and iOS Device sensors:** the accelerometer; Detecting shakes; Determining orientation; Responding to the accelerometer.
12. **Testing and debugging**

**Evaluation:**

* Coursework: 100%
  + 3 programming assignments (10, 10, 30%) 50%
  + One written supporting report for 30% programming assignments
  + Two quizzes 20%

**SWEN3001**

**ANDROID APPLICATION DEVELOPMENT I**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisites:**

COMP2171 - Object Oriented Design and Implementation **AND**

COMP3161 - Introduction to Database Management Systems

**Course Content:**

1. Android platform and architecture
2. Android user interface, layouts, views and GUI controls
3. Menus, Action Bar Menus, Toasts
4. Adapters, Dialogs, Intents
5. Storing and Retrieving Data: internal and external storage, preferences, SQLite Database
6. File Storage
7. Content Providers
8. Fragments
9. Developing for the Android marketplace
10. Java Programming: The Object class and its methods; Wrapper classes for primitive types; Inner and nested classes; The String, Stringbuffer and String Tokeniser classes, String processing; Handling files, input, output and serialisation, building database applications with JDBC; Localisation and Internationalisation, processing dates and time; Regular expressions; Exception handling and assertions; Multithreading and concurrency; Java collections framework; Graphical User Interface development using Swing; Java 5 features: enumerations, enhanced for loop, formatted output, Scanner autoboxing and unboxing of primitives, generic types, variable-length argument lists; JDK tools and deploying applications.

**Evaluation:**

* Coursework: 100%
  + A standalone Android application for 60%

the Android marketplace that uses the   
Android user interface, controls and local   
storage mechanisms

* + A supporting report for the 30%   
    Android application
  + An oral presentation 10%

**SWEN3002**

**ANDROID APPLICATION DEVELOPMENT II**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisite:**

SWEN3001 - Android Application Development I

**Course Content:**

1. Android Application Components: activities, broadcast receivers, services, notification manager
2. Mobile Web Applications: web apps overview, targeting screens from web apps, WebView, debugging web apps, best practices for web apps
3. Best Practices for Android Development: compatibility, supporting multiple screens, optimizing for other Android versions
4. Asynchronous Tasks: main UI thread, using AsyncTask
5. Accessing Remote Services: HTTP, DOM parsing, SAX parsing, JSON parsing, Android and distributed agent software systems
6. Server-side concepts
7. Client access to software agent system
8. Connectivity using, for example, Bluetooth, NFC, Wireless
9. Testing strategies

**Evaluation:**

* Coursework: 100%
  + Three (3) programming assignments 50%  
    (10%, 10%, 30%)
  + One individual report (critical 30%  
    appraisal of programming exercises)
  + Two (2) quizzes, 10% each 20%

**SWEN3003**

**WEB & MOBILE APPLICATION DEVELOPMENT I**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

SWEN1005 - Mobile Web Programming **AND**

COMP2171 - Object Oriented Design and Implementation **AND**

COMP3161 - Introduction to Database Management Systems

**Course Content:**

1. The Web
2. Web application architectures (e.g. MVC)
3. Interface design for web applications
4. Server-side components (e.g. Java servlets, Java Server Pages)
5. Manipulating a relational database from within a Java program, including PL-SQL and stored procedures
6. Session management
7. Scopes
8. Scope attributes
9. Request dispatching
10. Java application clients
11. Design patterns for web applications and data sources
12. Overview other frameworks (e.g., JavaServer Faces, Struts).

**Evaluation:**

* Coursework: 100%
  + A component-based Web application 50%   
    (Design and implement a component-based   
    Web application that provides dynamically   
    generated responses to user actions)
  + Supporting report 30%
  + Two quizzes (10% each) 20%

**SWEN3004**

**WEB & MOBILE APPLICATION DEVELOPMENT II**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisite:**

SWEN3003 - Web & Mobile Application Development I

**Course Content:**

1. The Android platform
2. Development environment for Android
3. Mobile application design
4. Interface design for mobile applications
5. Android software stack
6. Android application lifecycle
7. Activities & Intents
8. Services
9. Broadcast receivers
10. Content providers
11. SQLite database
12. On-phone resources: GPS, Telephony, Audio & video, Sensors, Connectivity
13. Business application development: an Android app as a rich client communicating with a server-side application

**Evaluation:**

* Coursework: 100%
  + A mobile Web application assignment 50%   
    (this application must interact with a   
    Web application)
  + A written supporting report 30%
  + Two quizzes (10% each) 20%

**SWEN3120**

**SOFTWARE ARCHITECTURE**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

COMP2140 - Software Engineering **AND**

COMP2171 - Object-Oriented Design and Implementation

**Course Content:**

1. **Software Architecture Concepts:** Architecture Trade-off Analysis Method (ATAM); Quality attribute trade-offs; Executing ATAM evaluation
2. **Architecture Design and Analysis:** Architectural Patterns and Tactics; Software architecture analysis concepts; Quality Attributes Workshop (QAW); Quality attribute scenarios; Attribute Driven Design (ADD)
3. **Architectural Documentation:** Principles of sound documentation; Using UML and other methods of documenting architecture; View types, styles and views; Choosing relevant views; Refinement; Interface documentation; Templates; Providing Justification for architecture to clients and developers (presentations and writing)
4. **Evaluating Software Architecture:** Architecture Trade-off Analysis Method (ATAM); Quality attribute trade-offs; Executing ATAM evaluation

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
  + Determine architectural drivers for 15%   
    a software-reliant system (group work)
  + Document the software architecture 15%   
    (group work)
  + Evaluate the software architecture 15%  
    (group work)
  + 2 In-course tests (1 hour each) (7.5% each) 15%

**SWEN3130**

**SOFTWARE PROJECT MANAGEMENT**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisite:**

COMP2140 - Software Engineering.

**Course Content:**

1. **The Role of Risk in the Software Life Cycle:** Risk categories including security, safety, market, financial, technology, people, quality, structure and process; Risk identification; Risk tolerance e.g., risk-adverse, risk-neutral, risk-seeking); Risk planning; Risk removal, reduction and control.
2. **Working in Teams:** Professional Ethics; Participation; Processes including responsibilities for tasks, meeting structure, and work schedule in a software team; Team Conflict Resolution; Virtual Teams (communication, perception, structure); Effort Estimation (at the personal level); Team Management including organisation, decision-making, role identification and assignment, individual and team performance assessment.
3. **Project Management:** Scheduling and Tracking; Project Management Tools; Cost/Benefit Analysis; Software Measurement and Estimation Techniques; Configuration Management and Version Control; Principles of Risk Management.

**Evaluation:**

* Final Examination (2 hours) 60%
* Coursework: 40%
  + Group Assignments (20% each) 40%

**SWEN3145**

**SOFTWARE MODELLING**

(3 Credits) (Level 3) (Semester 1)

**Pre-requisites:**

COMP2140 - Software Engineering **AND**

COMP2171 - Object Oriented Design and Implementation.

**Course Content:**

Requirements Specification Document Development (Precisely Expressing Requirements); Information Modelling (Entity-Relationship Modelling, Class Diagrams); Behavioural Modelling (Structured Analysis, State Diagrams, Use Case Analysis, Interaction Diagrams, Failure Modes and Effects Analysis); Structure Modelling (Architectural); Domain Modelling (Domain Engineering Approaches); Functional Modelling (Component Diagrams).

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
* 2 Assignments 20%
* 1 Project 40%

**SWEN3165**

**SOFTWARE TESTING**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisites:**

COMP2140 -Software Engineering **AND**

COMP2171 -Object Oriented Design and Implementation

**Course Content:**

Managing the Testing Process, Testing Principles and Techniques (Unit, Integration, Systems, Acceptance; Testing Types (State Based, Regression, Configuration, Compatibility, Alpha, Beta, and Acceptance); Test Driven Development; Test Plan Development; Reporting, Tracking, and Analysis of Problems encountered during Development.

**Evaluation**

* Final Examination (2 hours) 40%
* Coursework: 60%
* 2 Assignments 20%
* 1 Project Report 40%

**SWEN3185**

**FORMAL METHODS AND SOFTWARE RELIABILITY**

(3 Credits) (Level 3) (Semester 2)

**Pre-requisite:**

COMP2201 -Discrete Mathematics for Computer Science.

**Course Content:**

Role of Formal Specification and Analysis Techniques in the Software Development Cycle; Software Reliability Engineering Concepts and Practices; Software Reliability Models; Introduction to Mathematical Models and Specification Languages (Alloy, Z, VDM); Pre and Post Conditions, Invariants; Formal Approaches to Software Modelling and Analysis (Model Checkers, Model Finders); Tools in Support of Formal Methods.

**Evaluation:**

* Final Examination (2 hours) 40%
* Coursework: 60%
  + 2 Assignments 20%
  + 1 Project 40%

**SWEN3920**

**CAPSTONE PROJECT (SOFTWARE ENGINEERING)**

(6 Credits) (Level 3) (Semester 1, 2 and 3)

**Pre-requisites:**

COMP2140 - Software Engineering,

SWEN3130 - Software Project Management **AND**

SWEN3145 - Software Modelling.

**Co-requisite:**

SWEN3165 - Software Testing **AND**

SWEN3185 - Formal Methods and Software Reliability.

**Course Description:**

This course is the required group project course for all students majoring in software engineering. It is intended to be a capstone course that will bring together many of the topics that were covered in the rest of the curriculum. For this reason, students will be expected to take this course in their final year, for a period of six months beginning in semester two and ending in semester three. The project must encompass all matters relating to the software engineering process: requirements, design, coding, working in teams and project management.

**Evaluation:**

* Presentation and Demonstration of Final Product 10%
* Project Management Charter and Plan 15%
* Architecture and Design 15%
* Software Requirements Specification 30%
* Software Artefacts 30%

**SWEN4001**

**ADVANCED DATABASE SYSTEMS**

(3 Credits) (Level 4) (Semester 2)

**Pre-requisite:**

COMP3161- Introduction to Database Management Systems

**Course Content:**

1. Advanced database architectures, N-Tier, Grid Computing, Distributed Databases
2. Data Models, Relational and Object-Relational technologies, query languages including advanced SQL and Object SQL
3. Advanced Design and design issues; database development and performance
4. Current trends in Database development, including knowledge management, web and mobile databases; database issues for complex data including forensic and biometric data
5. Data mining
6. Analytics

**Evaluation:**

* Coursework: 100%
  + Two assignments (10% and 40%) 50%
  + One research paper (future 30%   
    trends of database technologies)
  + Two quizzes (10% each) 20%

**SWEN4002**

**I.T. CERTIFICATION I**

(3 Credits) (Level 4) (Semester 1)

**Pre-requisite:**

None

**Course Content:**

The course content will depend upon the specific certification/course pursued.

**Evaluation:**

The course assessment methods will be determined by the specific certification body.

****

#### AWARDS, PRIZES AND BURSARIES

##### Department of Computing

* **The Karl Robinson Award in Computer Science**

The Karl Robinson Award is a tribute to the life and work of the late Karl Robinson who distinguished himself as an invaluable member of the then Department of Mathematics & Computer Science. This award is presented to a final year student with the best academic performance in Computer Science. The winner of this award is the student with the highest average in first year, second year and Semester I of the third year Computer Science courses. In case of a tie, the award will be split equally among the winners.

* **NCB Best 2nd Year Computer Science/Software Engineering Award**

The National Commercial Bank Jamaica Ltd. celebrates the achievement of excellence in a field of study that will directly impact the digital economy. The winner of this award is the student with the highest average in first year, and Semester I of the second year Computer Science/ Software Engineering courses.

* **NCB Best 2nd Year Information Technology Award**

The National Commercial Bank Jamaica Ltd. recognizes the accomplishments of future contributors to the ICT sector in Jamaica. The winner of this award is the student with the highest average in first year, and Semester I of the second year Information Technology Engineering courses.

* **The Ezra Mugisa Award**

The Ezra Mugisa Award was introduced in 2020/2021 by the Department of Computing in honour of Dr. Ezra Mugisa in recognition of his contribution to the Department and as a motivation to the students of Computer Science. This award is a tribute to the work of Ezra Mugisa who retired from the University in 2018. He distinguished himself as an invaluable member of the Department of Mathematics & Computer Science and later the Department of Computing. He joined the UWI, Mona Campus in 1982 as a Data Controller in the then Sub-Department of Computer Science and subsequently obtained his Ph.D. in Computer Science at Imperial College, University of London. Dr. Mugisa taught numerous courses at both the B.Sc. and M.Sc. levels and served a number of terms as Head of the Computer Science Section and later as Head of the Department of Computing. The award is presented to a second-year student with the highest average in first year and Semester I of the second year Computing courses.