

PROG C7Z23: Programming Principles

| Module Details | |
|--------------------------|--|
| Module Code: | PROG C7Z23 |
| Full Title: | Programming Principles APPROVED |
| Valid From:: | Semester 1 - 2019/20 (June 2019) |
| Language of Instruction: | English |
| Duration: | 2 Semesters |
| Credits:: | 10 |
| Module Owner:: | Tony McCarron |
| Departments: | Unknown |
| Module Description: | Students completing this module will be capable of using a problem-solving approach to design, build and test solutions to fundamental programming problems. |

| Module Learning Outcome | |
|--|--|
| On successful completion of this module the learner will be able to: | |
| # | Module Learning Outcome Description |
| MLO1 | Design, build and test programming solutions using appropriate fundamental programming constructs. |
| MLO2 | Use simple data structures such as arrays and array lists to solve problems. |
| MLO3 | Use abstraction and decomposition as techniques to create well structured solutions using both user-defined static methods and standard library methods. |
| MLO4 | Read and write from files using library classes. |
| MLO5 | Create programs with GUI interfaces. |
| Pre-requisite learning | |
| Module Recommendations <i>This is prior learning (or a practical skill) that is strongly recommended before enrolment in this module. You may enrol in this module if you have not acquired the recommended learning but you will have considerable difficulty in passing (i.e. achieving the learning outcomes of) the module. While the prior learning is expressed as named DkIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).</i> | |
| No recommendations listed | |

| Module Indicative Content | |
|--|---------|
| Programming Constructs Variables, types, expressions and assignment, Console based input-output, validation. | |
| Control Structures Conditional and iterative control structures | |
| Api Familiarity with the methods of library classes e.g. String, Math, Random etc. | |
| Methods Method definition, invocation, variable scope, parameter passing, return types, and method overloading | |
| Simple Data Structures Arrays and ArrayLists. | |
| Input-output Files : reading and writing, GUI based programming. | |
| Module Assessment | |
| Assessment Breakdown | % |
| Course Work | 100.00% |
| Module Special Regulation | |
| | |

Assessments

| Full Time On Campus | | | |
|---|-----------------------|-------------------------|-----------|
| Course Work | | | |
| Assessment Type | Continuous Assessment | % of Total Mark | 20 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Every Second Week | Learning Outcome | 1,2 |
| Duration in minutes | 0 | | |
| Assessment Description Formative assessment consisting of practical exercises designed to build upon the theoretical topics covered and to give practice in building programming solutions relevant to the current topic. Assessments will be both individual and pair based. | | | |
| Assessment Type | Class Test | % of Total Mark | 20 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Sem 1 End | Learning Outcome | 1,2 |
| Duration in minutes | 120 | | |
| Assessment Description A lab-based practical exam, summative in nature giving the student the opportunity demonstrate their problem solving ability and their knowledge of topics covered to this point. | | | |
| Assessment Type | Continuous Assessment | % of Total Mark | 20 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Every Second Week | Learning Outcome | 1,2,3,4,5 |
| Duration in minutes | 0 | | |
| Assessment Description Formative assessment consisting of practical exercises designed to build upon the theoretical topics covered and to give practice in building programming solutions relevant to the current topic. Assessments will be both individual and pair based. | | | |
| Assessment Type | Class Test | % of Total Mark | 40 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Sem 2 End | Learning Outcome | 1,2,3 |
| Duration in minutes | 120 | | |
| Assessment Description A lab-based practical exam, summative in nature giving the student the opportunity demonstrate their problem solving ability and their knowledge of topics covered in this module. | | | |
| No Project | | | |
| No Practical | | | |
| No Final Examination | | | |

| Part Time On Campus | | | |
|---|-----------------------|-------------------------|-----------|
| Course Work | | | |
| Assessment Type | Continuous Assessment | % of Total Mark | 20 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Every Second Week | Learning Outcome | 1,2 |
| Duration in minutes | 0 | | |
| Assessment Description Formative assessment consisting of practical exercises designed to build upon the theoretical topics covered and to give practice in building programming solutions relevant to the current topic. Assessments will be both individual and pair based. | | | |
| Assessment Type | Class Test | % of Total Mark | 20 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Sem 1 End | Learning Outcome | 1,2 |
| Duration in minutes | 120 | | |
| Assessment Description A lab-based practical exam, summative in nature giving the student the opportunity demonstrate their problem solving ability and their knowledge of topics covered to this point. | | | |
| Assessment Type | Continuous Assessment | % of Total Mark | 20 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Every Week | Learning Outcome | 1,2,3,4,5 |
| Duration in minutes | 0 | | |
| Assessment Description Formative assessment consisting of practical exercises designed to build upon the theoretical topics covered and to give practice in building programming solutions relevant to the current topic. Assessments will be both individual and pair based. | | | |
| Assessment Type | Class Test | % of Total Mark | 40 |
| Marks Out Of | 100 | Pass Mark | 40 |
| Timing | Sem 2 End | Learning Outcome | 1,2,3 |

| | |
|---|-----|
| Duration in minutes | 120 |
| Assessment Description A lab-based practical exam, summative in nature giving the student the opportunity demonstrate their problem solving ability and their knowledge of topics covered in this module. | |
| No Project | |
| No Practical | |
| No Final Examination | |
| Reassessment Requirement A repeat examination <i>Reassessment of this module will consist of a repeat examination. It is possible that there will also be a requirement to be reassessed in a coursework element.</i> | |

Module Workload

Workload: Full Time On Campus

| Workload Type | Contact Type | Workload Description | Frequency | Average Weekly Learner Workload | Hours |
|-------------------|--------------|---|------------|---------------------------------|-------|
| Practical | Contact | Three 2-hour lab-based classes with the delivery of new content integrated with practical implementation. | Every Week | 6.00 | 6 |
| Independent Study | Non Contact | Independent work both lecturer- and self-directed. | Every Week | 1.00 | 1 |
| Directed Reading | Non Contact | Investigation and directed learning from both books and online resources. | Every Week | 1.00 | 1 |
| | | | | Total Weekly Learner Workload | 8.00 |
| | | | | Total Weekly Contact Hours | 6.00 |

Workload: Part Time On Campus

| Workload Type | Contact Type | Workload Description | Frequency | Average Weekly Learner Workload | Hours |
|-------------------|--------------|---|------------|---------------------------------|-------|
| Practical | Contact | One 3-hour lab based class with the delivery of new content integrated with practical implementation. | Every Week | 3.00 | 3 |
| Independent Study | Non Contact | Independent work both lecturer- and self-directed. | Every Week | 3.00 | 3 |
| Directed Reading | Non Contact | Investigation and directed learning from both books and online resources. | Every Week | 2.00 | 2 |
| | | | | Total Weekly Learner Workload | 8.00 |
| | | | | Total Weekly Contact Hours | 3.00 |

Module Resources

Recommended Book Resources

Julie Anderson & Herve J. Franceschi. (2018), Java Illuminated, 5th Edition. Jones & Bartlett Learning, [ISBN: 9781284140996].

Stuart Reges & Marty Stepp. (2014), Building Java Programs : A back to basics approach., 3rd Edition. Pearson Education, [ISBN: 9780133449440].

This module does not have any article/paper resources

Other Resources

Website, CodingBat, Stanford, Nick Parlante,

<https://codingbat.com/java>

Website, Marty Stepp. (2019), Practice-It, University of Washington,

<https://practiceit.cs.washington.edu>

Website, 'Java API',

<https://docs.oracle.com>