

|   |  |  |               |
|---|--|--|---------------|
| <b>Module code</b>  | SS-1203                                |  |               |
| <b>Module Title</b>   | Programming Fundamentals 2             |  |               |
| <b>Degree/Diploma</b>   | Bachelor of Science (Computer Science) |  |               |
| <b>Type of Module</b>   | Major Core                             |  |               |
| <b>Modular Credits</b>  | 4                                      | <b>Total student Workload</b>  | 10 hours/week |
|   |  | <b>Contact hours</b>   | 4 hours/week  |
| <b>Prerequisite</b>   | SS-1201 Programming Fundamentals 1     |  |               |
| <b>Anti-requisite</b>   | None                                   |  |               |
| <b>Aims</b>   |  |  |               |
| This module enhances the student's programming knowledge in order to acquire skills needed to write practical applications.   |  |  |               |
| <b>Learning Outcomes</b>  |  |  |               |
| <i>On successful completion of this module, a student will be expected to be able to:</i>   |  |  |               |
| Lower order :   | 20%                                    | <ul style="list-style-type: none"> <li>- describe abstraction and modularisation</li> <li>- differentiate the different types of dynamic data structures</li> </ul>  |               |
| Middle order :  | 60%                                    | <ul style="list-style-type: none"> <li>- implement classes using abstraction and inheritance</li> <li>- use library modules through their published API</li> </ul>   |               |
| Higher order:   | 20%                                    | <ul style="list-style-type: none"> <li>- design and implement well-structured library modules</li> <li>- design and implement well-structured GUI applications</li> <li>- apply event handling in the functions</li> </ul> |               |
| <b>Module Contents</b>  |  |  |               |
| <ul style="list-style-type: none"> <li>- Persistent data: writing, reading</li> <li>- Dynamic data structure: linked list, insertion, traversal, deletion</li> <li>- Module structure: composition, inheritance, polymorphism</li> <li>- Common design patterns: singletons, factories, composites, wrappers</li> <li>- Event handling: event loop, delegation models, publisher-subscriber, MVC</li> <li>- Graphical user interface</li> </ul> |  |  |               |
| <b>Assessment</b>   | Formative assessment                   | Interactive Quizzes and Feedback   |               |
|   | Summative assessment                   | Examination: 50%<br>Coursework: 50% <ul style="list-style-type: none"> <li>- 2 class tests (20%)</li> <li>- 1 written assignment (15%)</li> <li>- 1 laboratory exercise (15%)</li> </ul>                                   |               |