

**MODULE SPECIFICATION**

|  |
| --- |
| **Part 1: Information** |
| **Module Title** | Programming |
| **Module Code** | CY103 | **Level** | 4  |
| **For implementation from** | September 2020  |
| **UWE Credit Rating** | 30 | **ECTS Credit Rating** | 15 |
| **Faculty** | Environment and Technology | **Field** |  |
| **Department** | Computer Science and Creative Technologies |
| **Contributes towards**  | BSc (Hons)Cyber Security Technical Professional Compulsory |
| **Module type:**  | Standard  |
| **Pre-requisites**  | None |
| **Excluded Combinations**  | None  |
| **Co- requisites**  | None  |
| **Module Entry requirements** | None |
|  |
| **Part 2: Description**  |
| This module introduces students to the core concepts of programming with an introduction to algorithms and the characteristics of programming paradigms. Appropriate programming languages will be chosen to illustrate the concepts. It is not the intent of this module to teach apprentices to become proficient programmers. If this is a requirement of their employment, then it will be the employer’s responsibility. Among the topics included in this module are: introduction to algorithms, procedural, object-orientated & event-driven programming, the integrated development environment and the debugging process. Development lifecycles and processes are covered along with code repositories and version management. Lecture sessions cover the technical knowledge required. Designated practical work is included to ensure that apprentices have absorbed and understood the key principles involved.This module will be based on ensuring that apprentice’s practical skills and knowledge gained in the block release sessions are carried into the workplace to inform their employment and generation of evidence of competency. You will cover:* algorithms and program design
* fundamental programming concepts
* fundamental data structures
* typical program development environment and methods
* object-oriented programming
* functional programming
* event driven and reactive programming
* language translation and execution
* syntax analysis
* compiler semantic analysis;
* code generation
* coding in assembly language
* machine code
* scripting languages
* database query language
* the different aspects of the software development lifecycle and how they combine to deliver successful outcome, for example:need, design, trade-offs, implementation, deployment, support, evolution, validation, verification and assurance
* different approaches to developing software, including sequential, iterative/agile, etc.
* advantages and disadvantages of different software development processes along with choice of process in different contexts.
* selection and use of different tools and environments that support software development at different stages in the lifecycle
* the principles of systems engineering, including all aspects of technology, people, culture and process and the environment within which a system of interest exists and operates
* the benefits of a system approach to dealing with challenges arising from complexity, emergence, adaption and co-evolution
 |
| **Part 3: Assessment**  |
| This module is assessed by two methods: practical coursework and an examination (2 hours)Component A: Unseen examination (2 hours)Apprentices will be assessed on their knowledge and application of development lifecycles, methodologies and processes. Component B: Practical coursework (fully documented)Apprentices will design algorithms and implement a 4-function calculator as an assembler program, a GUI desktop application and a cloud-based solution. In the desktop and cloud solutions calculations must be stored and retrieved in a suitable file system or database. In each case there must be a technical description, code and evidence of testing and correct function. A short report will explain the development methodologies used in each case.  |
|  |
| Identify final timetabled piece of assessment (component and element) | **Component B1** |
| **% weighting between components A and B** (Standard modules only) | **A:**  | **B**:  |
| **40%** | **60%** |
|  |
| **First Sit** |
| **Component A** (controlled conditions)**Description of each element** | **Element weighting****(as % of component)** |
| 1. Unseen examination (2 hours) | 100% |
| **Component B** **Description of each element** | **Element weighting****(as % of component)** |
| 1. Practical coursework (fully documented) | 100% |
| **Resit (further attendance at taught classes is not required)** |
| **Component A** (controlled conditions)**Description of each element** | **Element weighting(as % of component)** |
| 1. Unseen examination (2 hours) | 100% |
| **Component B Description of each element** | **Element weighting(as % of component)** |
| 1. Practical coursework (fully documented) | 100% |
|  |
| **Part 4: Learning Outcomes & KIS Data** |
| **Learning Outcomes** | On successful completion of this module students will be able to:Write, test, and debug programs in high and low level languages and scripts. (Component B)Apply system engineering and software development methodologies and models. (Component A and B)Design and implement algorithms in different languages within a suitable Integrated Development Environment (IDE). (Component B)Explain and compare development lifecycles and processes. (Component A) |
| **Key Information Sets Information (KIS)****Contact Hours****Total Assessment** |  The table below indicates as a percentage the total assessment of the module which constitutes a;**Written Exam**: Unseen or open book written exam**Coursework**: Written assignment or essay, report, dissertation, portfolio, project or in class test **Practical Exam**: Oral Assessment and/or presentation, practical skills assessment, practical exam (i.e. an exam determining mastery of a technique)  |
| **Reading List** | Reading list to be added |

***FOR OFFICE USE ONLY***

|  |  |
| --- | --- |
| **First Approval Date (and panel type)** | *Date of first {panel} approval*  |
| **Revision ASQC Approval Date** *Update this row each time a change goes to ASQC* |  | **Version**  | *1* | *Link to RIA*  |
|  |  |  |
|  |  |  |
|  |  |  |