Module: Programming 361

Module name:	Programming 361
Code:	PRG361
NQF level:	6
Type:	Speciality – Diploma in Information Technology (Programming)
Contact Time:	84 hours
Structured time:	14 hours
Self-directed time:	42 hours
Notional hours:	140 hours
Credits:	14
Prerequisites:	PRG262

Purpose

The main focus of this module is on providing a comprehensive foundation sufficient for students to create new and/or modify existing applications to meet enterprise real-world requirements. The module brings together all the concepts learnt in the pre-requisite programming offerings and adds more advanced topics that blends to allow learning application of various technologies required to build enterprise applications. It addresses application programming interfaces, architectural choices, multi-threading, sockets programming, design patterns, and advanced programming practices to enable distribution, integration and security of desktop enterprise applications.

Outcomes

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

- An understanding of integrated knowledge of programming techniques and concepts as contested to construct computing systems using tools and services to develop computing systems that consider platform constraints, supports version control, tracks requirements and bugs, and automates building.
- The ability to identify, analyse, evaluate, critically reflect on and address complex problems, applying evidence-based solutions and theory-driven arguments through the use of application programming interfaces and frameworks when implementing solutions.
- An understanding of a range of methods to construct multi-tiered applications, evaluate and verbalise the value of using the different levels of logic separation.
- The ability to develop and communicate a solid understanding of the more advanced concepts of programming. Topics include data structures, reflection and design patterns and principles, and RESTful API implementation.
- The ability to take full responsibility for their own work, decision-making and use of resources to solve problems in unfamiliar and variable contexts exposed by different technologies and methodologies for tasks and be able to judge the relative merits of these to choose between the alternatives.

- The ability to manage processes in unfamiliar and variable contexts through the use of tools and services to develop computing systems that consider platform constraints, automates building, supports version control, tracks requirements and bugs.

Assessment

- Continuous evaluation of theoretical work through a formative and a summative test.
- Continuous evaluation of two projects, whereby the student design, implement and present on the outcome of the programming concepts for a given problem context.
- Final assessment through a written examination.

Teaching and Learning

Prescribed books (EBSCO)

- 🕮 ***Sean Burns (2019)*** **Hands-On Network Programming with C# and .NET Core : Build Robust Network Applications with C# and .NET Core*. Birmingham: Packt Publishing.***
- 🕮 ***Harihara Subramanian and Pethuru Raj (2019)*** **Hands-On RESTful API Design Patterns and Best Practices : Design, Develop, and Deploy Highly Adaptable, Scalable, and Secure RESTful Web APIs*. Birmingham, UK: Packt Publishing.***

Learning activities

Learning will be facilitated by the lecturer with student centred activities that involve problem-based learning where pupils are presented with challenges that replicate the situation in the real-world environment. This will be achieved through a combination between presentation of theoretical concepts, guided exercises, group work and discussions during the module.

Notional learning hours

Contact	Distance	Other	Type of learning activities	% Learning
y	y	n	Lectures (face-to-face, limited interaction or technologically mediated)	40%
y	y	n	Tutorials: individual groups	20%
n	y	n	Syndicate groups	10%
n	y	n	Independent self-study of standard texts and references (study guides, books, journal articles)	10%
n	y	n	Independent self-study of specially prepared materials (case studies, multi-media, etc.	20%

Syllabus

- Custom classes that implement generics
- Serialization and deserialization with generics and sockets
- Synchronization concepts on distributed desktop application
- Architectural choices of building distributed application solutions
- Concepts of design patterns and anti-patterns in very specific detail
- API (REST / Gateways)