Module: Programming 381

Module name:	Programming 381
Code:	PRG381
NQF level:	7
Type:	Fundamental – Bachelor of Computing (Software Engineering stream)
Contact Time:	38 hours
Structured time:	6 hours
Self-directed time:	46 hours
Notional hours:	90 hours
Credits:	9
Prerequisites:	PRG282

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.
- 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.
- The assignments or projects collectively will count 30% of your class mark.
- All tests will collectively account for 70% of your class mark.
- Your class mark contributes 30% towards your final mark for the subject, while the final assessment accounts for 70% of your final mark.

Teaching and Learning

Learning materials

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.

Additional Reference Material:

- Horstmann, S. C. (2013). *Core Java Volume II: Advanced Features*, 11th edition. [ISBN-13: 978-0-13-708189-9]
- All IT eBooks. (2018). Think Java - PDF eBook Free Download. [online] Available at: http://www.allitebooks.in/think-java/ [Accessed 13 Jun. 2018].
- Oracle.com. (2018). Java SE Tutorial Downloads. [online] Available at: http://www.oracle.com/technetwork/java/javase/java-tutorial-downloads-2005894.html [Accessed 13 Jun. 2018].

Learning activities

The teaching approach consists of a combination of formal lectures on theoretical and practical concepts, solving enterprise real-world problems through exercises and demonstrations of solutions in specific contexts. It is dialogue-oriented with a practical approach with mandatory projects written examinations, formative and summative assessments that must be completed during the module.

Notional learning hours

Activity	Units	Contact Time	Structured Time	Self-Directed Time
Lecture		27.0		14.0
Formative feedback		6.0		
Project	2	5.0		12.0
Assignment	1			3.0
Test	2		4.0	8.0
Exam	1		2.0	9.0

38.0	6.0	46.0

Syllabus

- Custom classes that implement generics
- Serialization and deserialization with generics and sockets
- Synchronization concepts on distributed desktop application
- Implement and control threads using the thread class
- Architectural choices of building distributed application solutions
- Concepts of design patterns and anti-patterns in very specific detail