

Module outline

Qualification	Faculty of Information Technology Bachelor of Science in Information Technology		
Module code	ITJA211	Module name	Programming in Java
NQF level	6	Credit value	12
Semester	1/2017	Year	2
Module lead	Marwick Makwindi	Internal moderator	Ndai Makhurane
Lecturing hours	52 (04 hours a week)	Tutorial hours	N/A
Notional hours	120	Copy-editor	Isobel Coetzee

The module outline must be read in conjunction with the study guide and the prescribed textbook (if applicable). This document will be the first port of call in understanding what will be assessed and which assessments form part of the module.

The purpose of the module outline is to highlight:

- The learning outcomes and assessment criteria that need to be met to pass the module
- The assessment required to be completed for the module
- The additional resources required for the module
- The topics that will be focused on for the module

Module aim

This module's aim is to provide students with an understanding of the principles of programming in Java, exploring the object oriented nature of the language and the multi-platform versatility offered.

Module description

Object oriented programming is an industry-proven method for developing reliable modular programs and is popular in software engineering and systems development. Consistent use of object oriented techniques can lead to shorter development lifecycles, increased productivity, adaptable code, reuse of different technologies, the interaction of different systems using common platforms and therefore lower the cost of producing and maintaining systems.

Java is synonymous with the object orient paradigm offering all the features of the technology in a format that can be used on many differing systems. The development of systems with Java objects simplifies the task of creating and maintaining complex applications.



Many environments use Java as its 'underpinning' framework, with Java applications found on mobile phones, dedicated systems, web-based multimedia, security and control systems as well as traditional applications and bespoke operating systems.


The module is not specific to one instance of the Java programming language and may be used to deploy, among others, mobile applications, bespoke applications or web-based solutions.


Learning outcomes


By the end of this module, students will be able to:

Learning outcomes	Assessment criteria
1. Explain the principles of programming in Java.	1.1 Discuss the principles, characteristics and features of programming in Java. 1.2 Critically evaluate the environmental flexibility of programming in Java.
2. Design Java solutions.	2.1 Design a Java programming solution to a given problem. 2.2 Explain the components and data and file structures required to implement a given design.
3. Implement Java solutions.	3.1 Implement a Java programming solution based on a prepared design. 3.2 Define relationships between objects to implement design requirements. 3.3 Implement object behaviours using control structures to meet the design algorithms. 3.4 Identify and implement opportunities for error handling and reporting. 3.5 Make effective use of an Integrated Development Environment (IDE) including code and screen templates.
4. Test and document Java solutions.	4.1 Critically review and test a Java programming solution. 4.2 Analyse actual test results against expected results to identify discrepancies. 4.3 Evaluate independent feedback on a developed Java program solution and make recommendations for improvements. 4.4 Create user documentation for the developed Java program solution. 4.5 Create technical documentation for the support and maintenance of a Java program solution.

Prescribed resources	
	Study guide Mapundu, M., Snyman, S. 2015. Programming in Java. Johannesburg: CTI EDUCATION GROUP.
	Textbook/e-book Deitel, P., Deitel, H. 2015. Java, How to program: Late Objects, Global Edition. 10th edition. New Jersey: Pearson Education. ISBN: 9781292070018
	The following resources will be made available on your learning management system: <ul style="list-style-type: none"> • Study guide • Assignment specification • Module outline • Assignment resources

Recommended resources	
	<p>The study guide provides students with the essentials to understand the learning outcomes. Take note, however, that Information Technology is a fast-developing discipline and textbooks are frequently updated; students should, therefore, use the latest editions, where available. Recommended resources should be used for reference purposes when conducting research for assignments. There is a range of general resources related to this module, including the following:</p> <p>Textbook/s or e-book/s</p> <p>Bloch, J. 2008. <i>Effective Java. Second Edition</i>. New Jersey: Prentice Hall.</p> <p>Goetz, B. 2006. <i>Java Concurrency in Practice</i>. Boston: Addison Wesley.</p> <p>Niemeyer, P. 2005. <i>Learning Java, Third Edition</i>. Sebastopol: O'Reilly.</p> <p>Schmuller, J. 2004. <i>Sams teach yourself UML in 24 hours. Third Edition</i>. Indiana: Sams Publishing.</p> <p>Online document/s</p> <p>Cadenhead, R. 2012. <i>Sams Teach Yourself Java in 24 hours</i>. [Online] Available at: https://progwolf.files.wordpress.com/2013/04/java-7-android-sams-teach-yourself-in-24-hours.pdf [Accessed: 9 September 2015].</p> <p>Website/s</p> <ul style="list-style-type: none"> • Web pages provide access to a further range of Internet information sources. • Lecturers may download the web-related material for students to access offline. • Students must use this resource with care, justifying the use of information gathered. <p>http://java.sun.com/docs/books/tutorial/</p> <p>http://math.hws.edu/javanotes/</p>

Recommended resources	
	http://www.codecademy.com/ www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml

Supporting documents	
	<p>Dietrichsen, P. & Bester, R. 2017. <i>Academic Skills</i>. Johannesburg: CTI Education Group.</p> <p>Geyer, L. 2015. <i>Harvard Referencing at CTI Education Group</i>. Johannesburg: CTI Education Group.</p>

Essential requirements

- Access to a resource centre or a library with a wide range of relevant resources, including: textbooks and e-books, newspaper articles, journal articles, organisational publications, databases, etc.
- Access to a range of academic journals in electronic format via EBSCOhost or other databases
- Whilst some procedural languages are commercially available, there are also free languages available incorporating a diverse range of commands, commonly deployed on many platforms. Campuses must ensure that in the case of mobile platforms, the applicable free emulators are available.
- Students must have access to facilities, which allow them the opportunity to fully evidence all of the criteria of the module.
- Students must develop an application that may be event driven, an applet, or command line driven and it may work on a range of platforms. It may be web based, GUI based, a games console or a deliverable for a mobile platform amongst many other solutions.
- Campuses must use a range of design methodologies, ensuring that the method selected is suited to the environment selected as well as the programming language of choice. Implementation must be based on a suitably structured problem that ensures the use of Java elements. Campuses must select a programming activity, or use an external source (employer, commissioner, open source).
- Working with a local programming-based organisation or using Internet-based open source projects would enhance the students' experience and offer a relevant vocational context.

Employer engagement and vocational contexts

- A greater emphasis is placed on practical application and as far as possible lecturers should make the work applicable to real world problems and situations.
- Where the lecturer sees appropriate opportunities, it is suggested that they make use of guest lecturers. This should be arranged according to the policies and guidelines followed on campus.

ICT requirements

ICT required	Reason	Lecture week/s
Computer with Java IDE, MS Office Access and MySQL2010	Lab work	2 - 12

All lab work must be completed on desktop computers.

Assessment details

Methods of assessment	Weighting ¹	Dates
Assignment	20%	Due date: 27/03/17 - 31/03/17
		Defined submission date will be stipulated by the module lecturer.
		Scope of coverage: Weeks 1 - 6
Semester test	20%	Scheduled week: 18/04/17 - 26/04/17
		Scope of coverage: Weeks 5 - 9
Continuous assessment	10%	<ul style="list-style-type: none"> The assignment and semester test do not cover weeks 10, 11 and 12. Lecturers may thus try to ensure that these weeks are covered within continuous assessment and the examination as set by the lecturers. A minimum of one activity per quarter. These assessments are compulsory.
		Scope of coverage: Weeks 1 - 11
		Marks captured: 16 May 2017
Initial examination	50%	29/05/17 - 09/06/17
Supplementary examination	50%	03/07/17 - 10/07/17
Special examination	100%	26/07/17 - 26/07/17

Due Performance (DP)

Students are required to meet a DP minimum in order to qualify to write their examinations. The DP requirement for all modules is a minimum mark of 40% for their coursework, which includes assignments and tests.

This means that a student will only be allowed to attempt an examination, if the final coursework average is 40% or higher. If the student does not meet the DP requirement, he/she will not be allowed to attempt the examination and will have to repeat the module.

The DP requirement is calculated as follows:

- $[(\text{Continuous assessment mark} \times 0.10) + (\text{Semester test mark} \times 0.20) + (\text{Assignment mark} \times 0.20)] \times 2$

In order to pass the module, a sub-minimum mark of 40% or higher is required for the examination and a final average of 50% or higher is required for the entire module.

¹ Refer to the **CTI Conditions of Enrolment**, available on myLMS.

Putting together a portfolio of evidence

Students must demonstrate, through the presentation of evidence, that they have met all module requirements within the qualification being undertaken. To do this, they must organise their evidence into what is known as a 'portfolio'.

A portfolio will take time and effort to complete; it is also a means of focusing and demonstrating student strengths and achievements to others. A portfolio is thus an important resource that many students may find useful to retain once they have achieved their qualification, particularly when applying for future positions.

Students are encouraged to read more into building their portfolio and to begin populating their evidence to illustrate their full skill-set to future employers.

Module content

Students are required to attend all classes; in addition, exercises and activities, which are supplied by lecturers, are compulsory.

Continuous assessments may run throughout the semester.

Lecture weeks	Topics and assessment criteria covered	Assessments	Study guide and textbook references
1 06/02/17 - 10/02/17	<ul style="list-style-type: none"> Introduction to Java Environment AC: 1.2, 2.1, 2.2, 3.1, 3.2 		<ul style="list-style-type: none"> Unit 1 Chapter 1, 2, 5, 7, 8
2 13/02/17 - 17/02/17	<ul style="list-style-type: none"> Introduction to Java Environment AC: 1.2, 2.1, 2.2, 3.1, 3.2 		<ul style="list-style-type: none"> Unit 1 Chapter 1, 2, 5, 7, 8
3 20/02/17 - 24/02/17	<ul style="list-style-type: none"> GUI components AC: 2.1, 3.1, 3.5 		<ul style="list-style-type: none"> Unit 2 Chapter 12, 15, 22
4 27/02/17 - 03/03/17	<ul style="list-style-type: none"> GUI components AC: 2.1, 3.1, 3.5 		<ul style="list-style-type: none"> Unit 2 Chapter 12, 15, 22
5 06/03/17 - 10/03/17	<ul style="list-style-type: none"> Control Statements and Relational Operators AC: 2.1, 2.2, 3.3 		<ul style="list-style-type: none"> Unit 3 Chapter 3, 4, 14
6 13/03/17 - 17/03/17	<ul style="list-style-type: none"> Control Statements and Relational Operators AC: 2.1, 2.2, 3.3 		<ul style="list-style-type: none"> Unit 3 Chapter 3, 4, 14
20/03/17 - 24/03/17	Semester break		
7 27/03/17 - 31/03/17	<ul style="list-style-type: none"> Arrays AC: 1.1, 2.1, 2.2, 3.1, 3.2 	Assignment submission ²	<ul style="list-style-type: none"> Unit 4 Chapter 6

² Date will be stipulated by the module lecturer.

Lecture weeks	Topics and assessment criteria covered	Assessments	Study guide and textbook references
8 03/04/17 - 07/04/17	<ul style="list-style-type: none"> Arrays AC: 1.1, 2.1, 2.2, 3.1, 3.2 		<ul style="list-style-type: none"> Unit 4 Chapter 6
9 10/04/17 - 13/04/17	<ul style="list-style-type: none"> Exception handling AC: 3.4 		<ul style="list-style-type: none"> Unit 5 Chapter 6, 11
Semester tests 18/04/17 - 26/04/17	No lectures	Semester tests	
10 02/05/17 - 05/05/17	<ul style="list-style-type: none"> Applets AC: 1.1, 1.2, 2.1, 3.1, 3.5 		<ul style="list-style-type: none"> Unit 6
11 08/05/17 - 12/05/17	<ul style="list-style-type: none"> Testing and Documentation AC: 4.1, 4.2, 4.3, 4.4, 4.5 		<ul style="list-style-type: none"> Unit 7
Sick tests 08/05/17 - 12/05/17	Lectures continue	Sick tests	
12 15/05/17 - 19/05/17	<ul style="list-style-type: none"> Unified Modelling Language for object-oriented design AC: 2.1, 3.1, 3.2 	All marks captured on e-Vision by 16 May 2017	<ul style="list-style-type: none"> Unit 8
Revision week 22/05/17 - 26/05/17	Revision and examination preparation		
29/05/17 - 09/06/17	Initial examination session		
03/07/17 - 10/07/17	Supplementary examination session		
26/07/17 - 26/07/17	Special examination		