

Study Guide

Programming 321

Academic Year 2018



Mrs. A. Joy

“Many people, other than the authors, contribute to the making of a book, from the first person who had the bright idea of alphabetic writing through the inventor of movable type to the lumberjacks who felled the trees that were pulped for its printing. It is not customary to acknowledge the trees themselves, though their commitment is total.”

Forsyth and Rada, *Machine Learning*



All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form by means electronic, mechanical, photocopying, recording or otherwise, without the written consent of the publisher.

Academic Year 2018

Version 2.1 (July 2017)

Contents

Introduction	5
Personnel contact details.....	5
Lecturer	5
Senior Lecturer	5
Dean: Faculty of Information Technology	5
Counselling Department	5
Counsellor	5
Counsellor	5
Counsellor	5
Learning Component	6
Overview of the qualification.....	6
Overview of subject	6
Purpose.....	6
Outcomes of subject in relation to other subjects and the qualification exit level outcomes	7
Application of critical cross field outcomes	7
Schedule.....	7
Summary	8
Detailed	8
Learning outcomes and assessments	9
Assessment	14
Assessment methods and activities.....	14
Assessment schedule.....	14
Predicate (class) mark.....	14
Promotion requirements	15
Work-integrated learning (WIL).....	15
Assessment administration	15
Resources	15
Mode of delivery	15
Quality assurance	15
Code of conduct.....	16
Theoretical classes.....	16
Practical sessions	16
Tests	16
Assignments& projects	16
Grievance procedure	17
Appendix A. Glossary of unfamiliar terms/concepts/acronyms	18
Appendix B. Procedures	19

Naming conventions for assignments	19
Upload procedure for assignments	19
Email Submission procedure for assignments and projects.....	20

Organisational Component

Introduction

Welcome to the subject Programming 321. The purpose of the specialisation is to mainly develop skilled people that can be successfully deployed within industries, including advanced proficiencies within the fields of application development. In this course the student will further their knowledge of Java programming by tackling more advanced programming concepts and newer more advanced technologies. The principles learnt in the prerequisite subjects will be applied in this subject as a new language is learnt and newer technologies are ventured in. It is an invitation to challenge you, in the attempt to gain new knowledge over and above that which is covered in the syllabus.

The student is encouraged to dedicate sufficient time to practice all the concepts covered in this subjects and to do extensive research and study.

Personnel contact details

Academic Department

Lecturer

Name: Mrs. A. Joy
E-Mail: Joy.A@belgiumcampus.ac.za

Senior Lecturer

Name: Mr. F. Venter
Email: Venter.f@belgiumcampus.ac.za

Dean: Faculty of Information Technology

Name: Mr. T. Kritzinger
Email: kritzinger.t@belgiumcampus.ac.za

Counselling Department

Counsellor

Name: Dr G. Steyn
Email: steyn.gg@belgiumcampus.ac.za

Counsellor

Name: Ms N. Roos
Email: Roos.n@belgiumcampus.ac.za

Counsellor

Name: Ms A. Narine
Email: narine.a@belgiumcampus.ac.za

Learning Component

Overview of the qualification

The purpose of the specialisation is to mainly develop skilled people that can be successfully deployed within industries, including advanced proficiencies within the fields of application development.

One of the first topics are the official coding conventions of Java, these are the conventions determined by Sun, now a part of Oracle. These coding conventions are used in most companies, though the conventions of some companies differ sometimes from the official ones. It is still important to learn how to code by these guidelines. From now on you will be expected to follow these coding conventions.

The second topic is generics; generics allow us to use collections, methods and classes that can handle multiple types of data. This allows us to re-use the code.

After this we will see how to transport text and objects using standard Input / Output and Serialization. This will be the basics of transporting data using Java. We will see examples of how we transport text and objects using streams, XML and JSON.

Then we will do a small recap on JDBC. Normally this subject should be handled of in Programming Java 200 but since this is one of the most important things in software development we will see a recap of this. This time we will also handle how to work with stored procedures and Java.

Chapter 8 will handle multithreading, multithreading allows us to make more complex applications. We will be able to run multiple parts of a program simultaneously and this will be of the highest importance in chapter 9, which handles networking in Java.

After we know how multithreading works we will start with networking in Java. Here we will see how to create distributed applications using sockets and RMI and what the advantages are towards other technologies.

Then we will see what Reflection is and how it is used. What its powers are and how we can use it to make more advanced applications.

The last chapter is dedicated to design principles and patterns. These allow us to code better and find easy solutions for re-occurring problems.

After this course you should have gained the qualities and knowledge to be a good Java developer that manages to build more advanced Java applications without help of others and in team.

Overview of subject

Purpose

The student is taught some of the more advanced subjects of java developing

- Generics
- Java I/O

- Serialization
- Java and XML
- Java and JSON
- Java Database Connectivity
- Multithreading
- Distributed applications in Java
- Reflection
- Design patterns and principles

The student will also have enough knowledge at the end of this subject to make considerations on what other technologies can be applied to create distributed applications.

Outcomes of subject in relation to other subjects and the qualification exit level outcomes

- Improved analytical thinking and problem solving skills.
- Convert object oriented designs expressed using diagrams and text into software solutions.
- Use advanced programming constructs to solve larger programming problems, including: Exceptions handling specific to distributed applications.
- Use object oriented libraries and frameworks when implementing a solution.
- Given a problem description for a single user program, students will be able to: Produce a design to address the problem using appropriate techniques, diagrams, and text.
- Use a contemporary integrated development environment to build and debug source code.

Application of critical cross field outcomes

- Throughout this course the student's ability to identify and solve problems in which responsible decisions will have to be made using critical and creative thinking. Troubleshooting of computer related problems and debugging code will contribute towards this.
- The student will grasp the concepts of Serialization and use these over a network. The student will see the different types to create a distribute application and will have a practice in this.
- Generics and reflections will be covered; this to see some of the more advanced techniques of programming so code can re-used and dynamic systems can be build.
- The student will experiment with multithreading and will use this in distributed applications.
- We will see how to combine java with XML and JSON to store data. We will use several parsers for this.
- There will be a short recap on Java Database Connectivity; here we will also cover stored procedures, something that wasn't covered in PJV 200.
- The student will learn how to solve a problem using a design pattern, learn the types of design patterns and how to implement these.

Schedule

Summary

Assessment:	Date
Test 1	2018-07-13
Test 2	2018-07-20
Test 3	2018-07-27
Test 4	2018-08-03
Test 5	2018-08-13
Summative Test	2018-08-17
Project 1	2018-07-18
Project 2	2018-08-01
Project 3	2018-08-14

Detailed

Day	Topics
1	Learning outcome 1
2	Learning outcome 2
3	Learning outcome 3
4	Learning outcome 4
5	Class Test 1
6	Learning outcome 5
7	Learning outcome 6
8	Learning outcome 7 Submission: Project 1
9	Learning outcome 8 Assessment Project 1
10	Class Test 2
11	Learning outcome 9
12	Learning outcome 10
13	Learning outcome 11
14	Learning outcome 12
15	Class Test 3
16	Learning outcome 13
17	Learning outcome 14
18	Learning outcome 15 Submission: Project 2
19	Learning outcome 16 Assessment Project 2
20	Class Test 4
21	Learning outcome 17
22	Learning outcome 18
23	Learning outcome 19

	Learning outcome 20
24	Public Holiday
25	Campus Closed
26	Class Test 5 Learning outcome 21
27	Learning outcome 22
28	Learning outcome 23 Submission: Project 3
29	Assessment Project 3
30	Summative Test

This schedule is subject to change.

Learning outcomes and assessments

Learning outcomes and Assessment 1

Learning outcomes 1	Assessment 1
General Java Programming principles <ul style="list-style-type: none"> Java Coding conventions Recap of prerequisites <ul style="list-style-type: none"> Access modifiers Abstract methods Concepts of abstract classes, methods Interfaces <ul style="list-style-type: none"> Concept of Interfaces and the need Methods and variables in an Interface Abstraction through interfaces and abstract classes Final and static keywords 	<ul style="list-style-type: none"> Understand the basic Java coding conventions covered in PJV100 and PJV200. Understand and make use of Classes and methods. Differentiate between interfaces and abstract classes. Apply relevant keywords to modify application behaviour

Learning outcomes and Assessment 2

Learning outcomes 2	Assessment 2
Recap of prerequisites <ul style="list-style-type: none"> Arrays Collections <ul style="list-style-type: none"> LISTS (ArrayList, LinkedList) SETS (HashSet, TreeSet) MAPS(HashMap, Hashtable, TreeMap) 	<ul style="list-style-type: none"> Discuss various Java collections and differentiate between them. Make decisions on best Java collections to apply given a scenario

--	--

Learning outcomes and Assessment 3

Learning outcomes 3	Assessment 3
<ul style="list-style-type: none"> Continue Collections <ul style="list-style-type: none"> QUEUE (PriorityQueue) Introduce Concepts of generics <ul style="list-style-type: none"> Generic collections Creating custom generic classes and methods 	<ul style="list-style-type: none"> Discuss the idea behind a generic implementation Create simple custom classes that implement generics

Learning outcomes and Assessment 4

Learning outcomes 4	Assessment 4
Recap Exceptions JDBC recap <ul style="list-style-type: none"> SQL statements and C.R.U.D operations Statements PreparedStatements Selecting and processing ResultSets Updating, Inserting, Deleting Executing batches	<ul style="list-style-type: none"> Apply best practice with exception handling Create database driven applications with the concepts taught in PJV200.

Learning outcomes and Assessment 5

Learning outcomes 5	Assessment 5
String Manipulation StringBuffer and Builder Java I/O introduction <ul style="list-style-type: none"> Java Stream I/O fundamentals System Class File Class and its methods FileWriter and FileReader BufferedWriter and BufferedReader Byte Streams and Char Streams	<ul style="list-style-type: none"> Recognise the advantages of StringBuffer over String Discuss the streams in Java I/O. Discuss byte streams vs. character streams Discuss stream chaining Make use of the File and system classes

Learning outcomes and Assessment 6

Learning outcomes 6	Assessment 6
Java I/O continued <ul style="list-style-type: none"> InputStream, OutputStream classes and Reader, Writer Classes FileInputStream and FileOutputStream BufferedInputStream and BufferedOutputStream Serialization <ul style="list-style-type: none"> Serialization concepts 	<ul style="list-style-type: none"> Discuss Stream classes vs. Writer classes vs. Reader Classes Discuss the concepts of buffered streams Implement the various streams stated in this outcome. Discuss the concept of serialization and its uses.

<ul style="list-style-type: none"> ObjectInputStream and ObjectOutputStream 	<ul style="list-style-type: none"> Implement serialization and deserialization Implement serialization and deserialization with generics
--	--

Learning outcomes and Assessment 7

Learning outcomes 7	Assessment 7
<p>Multithreading</p> <ul style="list-style-type: none"> Define a thread Instantiate a thread Start a thread Thread States <ul style="list-style-type: none"> Running Runnable Waiting Sleeping Blocking Dead Thread Priorities Join and Yield methods Synchronizing code/ thread Locks. Object Methods, Wait and Notify 	<ul style="list-style-type: none"> Discuss the concepts of multithreading. State advantages of threaded applications. Thread class vs. Runnable Interface Discuss synchronization concepts. <ul style="list-style-type: none"> Thread states. Locks. Thread deadlocks. Discuss the lifecycle of a thread Implement and control threads using the thread class

Learning outcomes and Assessment 8

Learning outcomes 8	Assessment 8
<p>Java and XML</p> <ul style="list-style-type: none"> XML file usage and structure DOMParser <p>Introduction to JSON</p> <ul style="list-style-type: none"> Mapping between JSON and Java entities Encoding and decoding JSON in Java 	<ul style="list-style-type: none"> Discuss the usage and structure of XML files Consume basic XML files using the DOMParser Discuss the idea behind the JSON schema Encode and decode JSON in java

Learning outcomes and Assessment 9

Learning outcomes 9	Assessment 9
<p>Networking introduction</p> <ul style="list-style-type: none"> What are distributed application all about <p>Socket programming</p> <ul style="list-style-type: none"> Principles of Socket programming Networks, Packet and Protocols principles Addressing concepts Client and Server architecture principles Relevant I/O streams 	<ul style="list-style-type: none"> Make use of sockets to create a basic client and server application Discuss the concepts behind socket programming Send serialized data between client and server over sockets

<ul style="list-style-type: none"> • Reading From and Writing to a socket • Writing the Server Side of a Socket 	
---	--

Learning outcomes and Assessment 10

Learning outcomes 10	Assessment 10
Socket programming continued <ul style="list-style-type: none"> • Send serialized data between client and server over sockets • Recap of cover concepts on socket programming • Examples and exercises 	<ul style="list-style-type: none"> • See Learning Outcome 9

Learning outcomes and Assessment 11

Learning outcomes 11	Assessment 11
Socket programming continued <ul style="list-style-type: none"> • TCP Sockets 	<ul style="list-style-type: none"> • Create a basic TCP Socket application • See Learning Outcome 8 and Learning Outcome 10

Learning outcomes and Assessment 12

Learning outcomes 12	Assessment 12
Socket programming continued <ul style="list-style-type: none"> • UDP Sockets 	<ul style="list-style-type: none"> • Discuss the difference between TCP and UDP sockets • Motivate the various scenarios where TCP and UDP sockets should be applied • Create a UDP socket based application

Learning outcomes and Assessment 13

Learning outcomes 13	Assessment 13
Introduction to Java Remote Method Invocation (RMI) <ul style="list-style-type: none"> • Features of RMI • The RMI architecture • Designing the Remote interface • Implementing the Remote interface 	<ul style="list-style-type: none"> • State and discuss the features of RMI • Differentiate RMI to other technologies such as CORBA and Sockets • Illustrate the architecture of an RMI application • Design and implement RMI remote interfaces • Host and consume RMI services client

Learning outcomes and Assessment 14

Learning outcomes 14	Assessment 14
Remote Methods Invocation continued	<ul style="list-style-type: none"> • See Learning Outcome 12

<ul style="list-style-type: none"> • Hosting the server application to respond to requests • Consuming an RMI server • Recap on RMI based applications 	
---	--

Learning outcomes and Assessment 15

Learning outcomes 15	Assessment 15
Revision <ul style="list-style-type: none"> • Review of concepts covered so far • Question and answer session 	<ul style="list-style-type: none"> • Revisit assessment of previously taught outcomes

Learning outcomes and Assessment 16

Learning outcomes 16	Assessment 16
Reflection <ul style="list-style-type: none"> • Reflection concepts/fundamentals • Use of reflection • Drawbacks of reflection • Reflection on classes 	<ul style="list-style-type: none"> • Discuss what reflection is and its practical applications • Implement reflection to inspect other code

Learning outcomes and Assessment 17

Learning outcomes 17	Assessment 17
Reflection continued <ul style="list-style-type: none"> • Reflection on class members • Practical examples 	<ul style="list-style-type: none"> • See learning outcome 15

Learning outcomes and Assessment 18

Learning outcomes 18	Assessment 18
Design patterns introduction <ul style="list-style-type: none"> • What are design patterns • What are the categories of design patterns • What are anti-patterns MVC Recap	<ul style="list-style-type: none"> • Discuss and contrast the concepts of design patterns and anti-patterns • Discuss the categories of design patterns • Discuss the covered design patterns in very specific detail

Learning outcomes and Assessment 19

Learning outcomes 19	Assessment 19
Design patterns <ul style="list-style-type: none"> • Creational Patterns and Scenarios <ul style="list-style-type: none"> ○ Singleton ○ Factory ○ Factory Method ○ Abstract Factory 	<ul style="list-style-type: none"> • Discuss and contrast the concepts of design patterns and anti-patterns • Discuss the categories of design patterns • Discuss the covered design patterns in very specific detail

Learning outcomes and Assessment 20

Learning outcomes 20	Assessment 20
Design patterns continued <ul style="list-style-type: none"> • Creational patterns and scenarios 	<ul style="list-style-type: none"> • See Learning Outcome 19

<ul style="list-style-type: none"> ○ Abstract Factory ○ Singleton ○ Builder 	
--	--

Learning outcomes and Assessment 21

Learning outcomes 21	Assessment 21
Design patterns continued <ul style="list-style-type: none"> • Structural patterns and case scenarios <ul style="list-style-type: none"> ○ Adapter ○ Decorator ○ Bridge 	<ul style="list-style-type: none"> • See Learning Outcome 20

Learning outcomes and Assessment 22

Learning outcomes 22	Assessment 22
Design patterns continued <ul style="list-style-type: none"> • Behavioural Patterns <ul style="list-style-type: none"> ○ Observer ○ Strategy ○ Command 	<ul style="list-style-type: none"> • See Learning Outcome 21

Learning outcomes and Assessment 23

Learning outcomes 23	Assessment 23
Design patterns continued <ul style="list-style-type: none"> • Recap of all categories and design patterns covered Recap and closure of subject Programming Java 321	<ul style="list-style-type: none"> • See Learning Outcome 19-21 • Revisit all other Learning Outcomes covered in this subject

Assessment

Assessment methods and activities

Various types of assessments are used to evaluate students:

- **Tests:** Tests written during the module. This will include a summative test at the end of the module.
- **Practical's:** Demo or hand in reports on practical sessions in the computer lab.
- **Practical tests:** Tests written during class time.
- **Examination:** Theoretical or practical assessment written during faculty examination period.

Assessment schedule

Assessment dates as shown in the Schedule module will be followed, unless specifically arranged by the lecturer. These changes will be announced well in advance.

Predicate (class) mark

The following list summarises all assessments planned for the module, as well as their contribution towards the final mark:

Assessment	Weight
------------	--------

Test 1	30
Test 2	30
Test 3	35
Test 4	35
Test 5	35
Summative Test	45
Sub Total	210
Project 1	30
Project 2	30
Project 2	30
Sub Total	90
Total	300

Predicate

The prerequisite predicate to write examination is 45%.

Final mark

The final mark will be a 30:70 combination of the predicate and the examination.

Promotion requirements

A final mark of 50% is necessary to pass the above mentioned subject.

It is each student's own responsibility to consult the official examination notice boards to find out whether he/she qualifies for a supplementary examination.

Work-integrated learning (WIL)

This subject requires no work-integrated learning.

Assessment administration

The subject PRG321 will be assessed by means of examinations.

Tests are scheduled during the module and other assessments will have to be taken to build up a predicate if required.

There will be a final exam at the end of the semester covering all the work of the module.

Resources

Programming 321 – IT without frontiers series (prescribed book)

It is also highly recommended that the student do further reading on the various topics covered in the subject. Consult other sources such as the Internet and Journals, and Publications.

Mode of delivery

The subject is presented as a six week module.

Theoretical presentations are supported by practical exercises in the computer laboratory.

Quality assurance

Evaluation forms will be completed at the end of the module to evaluate the subject, course material and lecturer.

Code of conduct

Theoretical classes

- Students should comply with the **85%** class attendance rule.
- Students may not be late for class.
- No eating or drinking is allowed in the classroom.
- Switch off cell phones during class times.
- Regularly refer to the notice boards for notices, test timetables, and other communication.
- Prepare for the work to be covered in class in advance.
- If the student is absent, it is his/her own responsibility to catch up on the work covered in class.

Practical sessions

- Each member must do his/her own preparation according to the instructions.
- The computers will be provided to each student with the accompanying software required to complete this course.
- The computer lab must be left neat and tidy when you leave. Chairs pushed in, keyboard and mouse neat in front of the computer screen.
- No eating and drinking is allowed in the computer lab.
- If a student misses a practical session, it is his/her own responsibility to complete the practical.

Tests

- All bags and books must be put away according to the invigilator's instructions.
- All cellular telephones must be switched off.
- All written answers must be done in pen. Work done in pencil will be seen as rough work and will not be marked. Diagrams may be done in pencil.
- Show all your calculations in the allocated areas.
- Diagrams and schematics will be marked as follows: Marks will be allocated for the correct components or blocks added to the diagram or schematic and marks will be deducted for each wrong connection of these components or blocks on the diagram or schematic or for any additional components or blocks unnecessarily added to the diagram or schematic.
- If a student misses a test due to illness or death of a family member, he/she must submit the sick note or copy of the death certificate within 7 days of his/her return.

Assignments& projects

- All students will get a personal student account on the Belgium campus website. All assignments and projects must be submitted through the student portal on the Belgium campus website.
- ***No late assignments will be accepted. See procedures***
- Everything handed in by the student must be his/her own work.
 - Plagiarism is the act of taking the writings of another person and passing them off as one's own (Encyclopedia Britannica). This includes copying from books, websites, published works, journals, other students, etc.

- Plagiarism is a criminal offence and Belgium Campus is strongly in opposition to any shape or form thereof. A student caught performing this, is subject to a possible hearing and suspension.

Grievance procedure

- A student may approach the student counsellors (Ms. N. Roos, Dr. G. Steyn or Ms. A. Narine). The student can book an appointment with them via Mr. P. Nkambule at the Academic office or via e-mail.
- If the student counsellors deem the nature of a grievance as such that the Dean need to handle it, they will book an appointment with him on behalf of the student.
- A student should use the available form (on the student's portal or at the Academic office) to report an academic related complaint. They can submit the form at the Academic office or with the student counsellors. The Dean or the student counsellors will follow up and find a solution (if the complaint is valid). They will give feedback to the student as soon as possible.
- A student may also approach the SRC representative in his/her class.

Appendix A. Glossary of unfamiliar terms/concepts/acronyms

Level	Refers to the level of the qualification on the NQF; can vary from 5 to 8.
Credits	Refers to the number of SAQA credits allocated (1 credit = 10 notional hours) to a unit standard or qualification.
Study field	Refers to the 12 organising fields of the NQF. These fields can be divided into sub-fields. Typical fields are Education, Training and Development, or Communication Studies and Languages, etc.
Self-directed learning	Refers to learners who have become independent and self-reliant regarding their learning.
Study methods	These may vary from subject to subject, but entails elements such as exploration, evaluation, and understanding of the subject via diagrams, mapping, dialogues, reflective thinking, literature study, self-directed study, case studies, simulations, etc.
Notional hours	Refers to the average time needed by the average student to master the learning outcomes of a study programme or qualification, including the amount of contact time, structured learning and assessment opportunities.
Module	Refers to a component of the full study course and nominally consists of a related combination of specific learning outcomes. Modules should contain stated outcomes, objectives, activities/exercises, resources and assessment opportunities. The combination of related outcomes can also form a registered unit standard on the NQF.
Learning unit	Refers to a smaller component of a module or unit standard and can consist of learning linked to a single outcome. Combinations of related learning units form a module.
WIL (Work-integrated learning)	Also known as Experiential Learning. Refers to learning acquired in workplace contexts, e.g. practice teaching for prospective teachers in schools for a certain period every year as part of the training programme.
Assessment criteria	Statements which set guidelines for the development of assessment tasks at learning programme level and which will determine whether a learner has been successful in achieving an outcome. The learner must prove the achievement of the outcome.

Appendix B. Procedures

Naming conventions for assignments

Any assignment that a student creates for electronic submission via the website must conform to the following naming convention. The reason for this is to ensure that your assignment is received correctly by your lecturer; this prevents any inconvenience for you as student and the lecturer.

- LecturerSurname_LecturerFirstname_SubjectCode_StudentSurname_StudentName_AsmntX.zip
- LecturerSurname_LecturerFirstname_SubjectCode_StudentSurname_StudentName_ProjectX.zip

Examples:

- Venter_Francois_PRG211_Doe_John_Asnmt1.zip
- Welgemoed_Andries_PRG211_Doe_John_Project1.zip

Please note the extension used for your submission is a “.zip” archive. If you are not familiar with this please ask your lecturer to assist you.

Upload procedure for assignments

Each student is provided with a username and password when registering with Belgium Campus. This will typically be “Surname. Initial” and the password will be the same. You can change your password as soon as you are able to log into your Belgium Campus portal.

Below you will find a guide to log onto your portal and submit assignments.

1. Go to the following website and type in the URL www.belgiumcampus.ac.za
2. If this was successful you should see the website as illustrated in image 1 below.
3. Click on the button “Login”
4. Enter your Username and Password.
5. Click login.
6. If successful you should see a screen that resembles Image 2 below, otherwise you have entered incorrect information. Try login in again else see the administration office to assist you.
7. You will notice on your portal all the active assignments that you have. Click on the selected assignment and you should be transferred to the upload page where you can select your file for upload.
8. Your lecturer will assist you on your first upload.

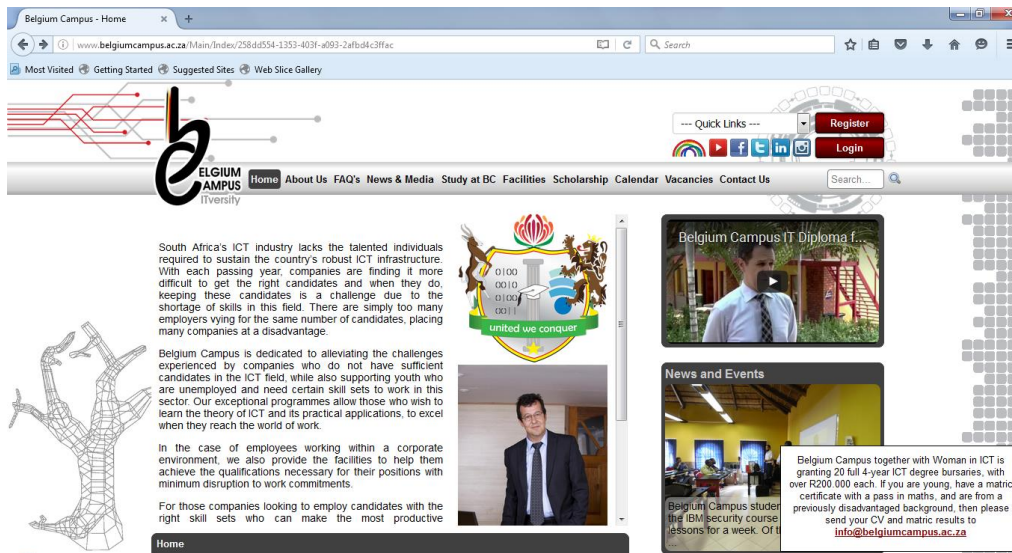


Image 1

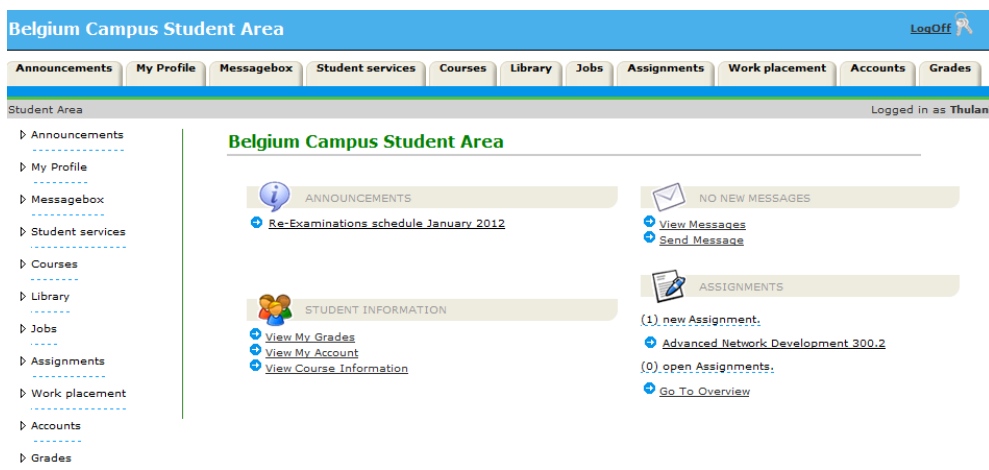


Image 2

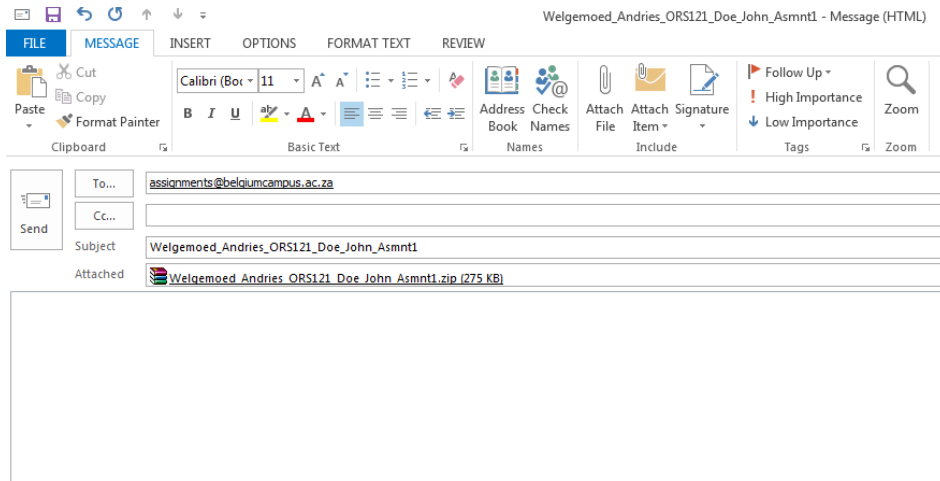
Email Submission procedure for assignments and projects

In the event that the Portal Submission procedure for assignments and projects fail due to technical difficulties, the student can use the Email Submission procedure for assignments and projects.

Please note this is only if the Portal Submission procedure for assignments and projects fail due to technical difficulties. **Late submission will not be accepted.**

Procedure:

1. Enter the address assignments@belgiumcampus.ac.za into the “To” field, as indicated in Image 3 below.
2. Enter
“LecturerSurname_LecturerFirstname_SubjectCode_StudentSurname_StudentName_AsmntX” into the “Subject” field. Note this should be the same as your submission file name, as indicated in Image 3 below.
3. Attached your submission file to the email, as indicated in Image 3 below.

*Image 3*