Tshwane University of Technology

We empower people

Faculty of Information and Communication Technology

Department of Computer Science

STUDY GUIDE

NQF LEVEL	NQF CREDITS	MODULE NAME	MODULE CODE	YEAR	SEMESTER
5	15	Principles of Programming B	PPB115D/PPBF15D/PPG115D/PPGF15D	2023	2

QUALIFICATION	SAQA ID:
Diploma in Computer Science	H/H16/E089CAN
Diploma in Multimedia Computing	H/H16/E090CAN
Diploma in Information Technology	H/H16/E088CAN
Diploma in informatics	H/H16/E061CAN

Compiled by Vuyisile Memani
Revised by Vathiswa Booi: July 2023

Contents

SECTION A – INTRODUCTION

1. Welcome

Welcome to **Principles of Programming B.** This module is a follow up to the Principles of Programming A module. In this module we further introduce the student to intermediate concepts of Object-Oriented Programming concepts (OOP) using the Java programming language. Some of the concepts to be learnt are modularisation; character manipulation; string manipulation; one-dimensional arrays; two-dimensional arrays; and user-defined classes. We are looking forward to working with you. Enjoy the module.

2. Introduction

The purpose of this study guide is to provide the learner with information needed that is crucial in completing this module. This study guide serves as a contract between the learner and the lecturer regarding what is to be done, and on which date it should be completed. All rules and regulations stipulated in this study guide should be aligned with the standard rules and regulations of Tshwane University of Technology. Any deviation from these standard rules and regulations will be regarded as invalid and must be reported to the HOD of the module's custodian department.

3. Name of the module

Principles of Programming B

4. Module Credits/Weight

15

5. Custodian department of this module

The custodian department of this module is the department of Computer Science.

6. Purpose of the module

This Principles of Programming B module is a 15 credit module with fundamental learning modules on NQF level 5 which prepares the student to apply programming and software engineering principles to provide solutions to a range of problems emanating in IT industry. This module builds on the knowledge and skills already obtained in "Principles of Programming A". The focus of this module is to introduce students to intermediate object oriented programming concepts in Java such modularisation; character manipulation; string manipulation; one-dimensional arrays; two-dimensional arrays; and user-defined classes. The student will be able to apply his/her knowledge of these intermediate programming concepts to the problems arising in the software industry.

This module forms part of the following qualifications:

- Diploma in Computer Science.
- Diploma in Informatics.
- Diploma in Information Technology.
- Diploma in Multimedia Computing

For the Diploma in Computer Science this module will individually contribute to achieving the exit level outcome of:

- ELO 1: Design and analyse algorithms and data structures to create efficient business solutions.

Therefore, upon completion of this module, the student should demonstrate knowledge, understanding and the ability to design small scale applications that deliver peace of mind to the users.

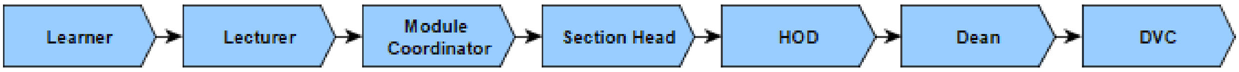
The intent of the module is to ensure that the students is well-rounded, and this can only be ensured if the individuals can interact effectively with their changing environment and act responsive and responsible within a variety of social and cultural settings. Aligned to the graduate attributes and ELO's students will be guided through activities like problem-solving and reflexive practices. In addition, students will acquire the following:

Knowledge: understand key terms, concepts, facts, principles, rules and theories.
Skills: select and apply standard methods, procedures and/or techniques.
Plan: manage and implement processes within a supported environment.

The module will require students to work individually and in groups thereby contributing towards both personal and professional ethics. As the students' progress in their study, they will be required to present their work to other fellow students, thereby developing their presentation and communication skills (skills identified as graduate attributes by the university). The technical knowledge obtained in this module together with the communication skills and presentation skill will prepare the students for the work place. This tie in with the instructional goal - To prepare diverse students for rewarding careers and responsible citizenry by providing a student-centred learning experience that is underpinned by a scholarship of teaching and learning.

7. Code of conduct

- Students may not be late for class. (Do not enter the class if you are late)
- Students should attend all classes. (The 85% class attendance rule applies)
- Cell phones should be switched off during class times.
- No eating or drinking in the classes.
- Students should strictly follow the schedule, unless otherwise stated. (page 4 of this study guide)
- Students should also look on the notice board outside the lecturer's office for updates on marks and communication.
- Each student should have a text book.
- Students should prepare for each class by researching the module.
- Students can see lecturer during consultation times, otherwise make an appointment with the lecturer for another time.
- Students should submit their sick note or death certificate within the specified time period.
- Since 2011 a student is not allowed to register more than three times for a module.
- All issues pertaining to the module must be reported to your lecturer. If the issue cannot be resolved adequately it must be escalated to the module coordinator. If the issue is still not resolved, it must be escalated to the section head. If the issue is still not resolved, it can be escalated to the HOD of the module's custodian department (see section A). Students may only report issues to the Dean of the faculty if the issue cannot be resolved by the HOD. Under no circumstances will issues be addressed if the proper reporting chain was not followed (see the figure below)



SECTION B – ORGANISATIONAL COMPONENT

8. Contact information of custodian department of this module

The custodian department of this module is the department of Computer Science.

Name & Surname	Campus	Office Location	Contact number and E-mail	Role in Programme
Me Molly Moche	Soshanguve South	20 – 103	012 382-9938 MollyMoche@tut.ac.za	Departmental administrator
Me Angelina Shonaphi Bhembe	E-Malahleni	14-G37	013 653-3165 BhembeAS@tut.ac.za	Administrator
Me Maesela Lebelo	Polokwane	1-G247	015 287-0757 LebeloM@tut.ac.za	Administrator

9. Contact information of lecturers

This module has one primary **module coordinator** that is responsible for coordinating the activities of the module. If this module is offered at one of the other learning sites and the primary module coordinator does not reside on the learning site, the activities of this module is managed by the primary module coordinator via a **site coordinato**r. There is one **facilitator** assigned to facilitate the teaching of each group.

9.1. Module coordinator/s

Module Coordinator	Vathiswa Booi
Office Location	20 – 214
Office Telephone Number	012 382 9061
Email address	BooiVM@tut.ac.za

9.2. Site coordinators

Soshanguve Site Coordinator	
Office Location	
Office Telephone Number	
Email address	

Emalahleni Site Coordinator	
Office Location	
Office Telephone Number	
Email address	

9.3. Facilitators

9.3.1 PPB115 Facilitators

Lecturer of Group 1	Booi Vathiswa
Office Location	20 – 214
Office Telephone Number	012 382 9061
Email address	BooiVM@tut.ac.za

Lecturer of Group 2	Marlo Liebenberg
Office Location	20 – 207
Office Telephone Number	012 382 9596
Email address	LiebenbergM@tut.ac.za

Lecturer of Group 3	Tshepo Raphiri
Office Location	12 -
Office Telephone Number	012 382 9197
Email address	RaphiriTS@tut.ac.za

Lecturer of Group 4	Vuyisile Memani
Office Location	12-138 (20-202)
Office Telephone Number	012 382 9749
Email address	MemaniV@tut.ac.za

9.3.2 PPBF15D Facilitators

Lecturer of Group 1	Zenzele Msiza
Office Location	20- G08
Office Telephone Number	012 982 9823
Email address	MsizaZA@tut.ac.za

Lecturer of Group 2	Marlo Liebenberg
Office Location	20 – 207
Office Telephone Number	012 382 9596
Email address	LiebenbergM@tut.ac.za

Lecturer of Group 3	Bongani Mabunda
Office Location	20-G11A
Office Telephone Number	012 382 9558
Email address	MabundaBT@tut.ac.za

Lecturer of Group 4	Vusi Msimango
Office Location	20 – G16
Office Telephone Number	012 382 9872
Email address	MsimangoVS@tut.ac.za

9.3.3 PPG115D Facilitators

Lecturer of Group 1	Wendy Birch
Office Location	20-209
Office Telephone Number	012 382 9053
Email address	BirchW@tut.ac.za

Lecturer of Group 2	Nokubongwa Zuma
Office Location	
Office Telephone Number	012 382 9012
Email address	ZumaNF@tut.ac.za

9.3.4 PPG115D Facilitators

Lecturer of Group 1	James Ramabu
Office Location	20-G18
Office Telephone Number	012 382 9837
Email address	RamabuTJ@tut.ac.za

Lecturer of Group 2	Solomon Odunaike
Office Location	
Office Telephone Number	012 382 9151
Email address	OdunaikeSA@tut.ac.za

Lecturer of Group 3	Calvin Phiri
Office Location	
Office Telephone Number	012 382 9558
Email address	PhiriMC@tut.ac.za

10. Library Contact Details

Soshanguve Faculty Librarian	Ms Rachel Raisibe Ntsoane
Office Location	Ground floor in library
Office Telephone Number	012 799 9509
Email address	NtsoaneRR@tut.ac.za

11. Student Support Contact Details

Student Support Services	Dr Shafeeka Dockrat
Office Location	Pta: Building 5, Room 5-705
Office Telephone Number	+27 12 382 4260
Email address	DockratS@tut.ac.za

12. Time table

The time table for class attendance is obtainable from the leaner management system (LMS).

13. Consultation

Consultation time slots will be displayed on the LMS. Consultations can be done online or face-to-face. Students must please contact the lecturer one day in advance via email and arrange a consultation session. Please remember to mention your name & surname, student number, module code and topic to be discussed during the consultation session.

14. Schedule

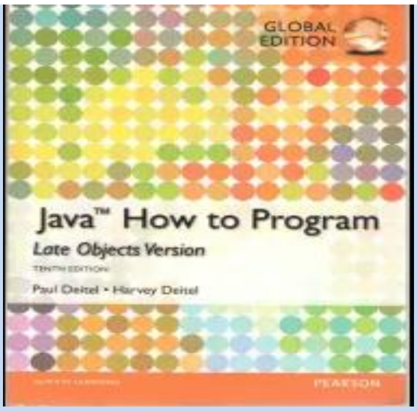
The schedule indicates all-important dates for activities such as, class activities, assignment due dates, class tests, excursions, practicals, project due dates, computer-based tests submission dates etc. Please ensure that you follow the schedule of your assigned group.

Weekday	Date	Activity / Specified Outcome to be addressed All Groups
Monday	10 Jul	Registration: Adding & Cancelling of Subject. Learning Outcome A.1.: Revision on Selection and Iteration control structures.
Tuesday	11 Jul	
Wednesday	12 Jul	
Thursday	13 Jul	
Friday	14 Jul	
Monday	17 Jul	Learning Outcome A.2: Revision on Iteration control structures. Learning Outcome B.1.1- B.1.1.6: Creating modularised interactive programs.
Tuesday	18 Jul	
Wednesday	19 Jul	
Thursday	20 Jul	
Friday	21 Jul	
Monday	24 Jul	Learning Outcome B: Creating modularised interactive programs (cont.) **More Exercises on learning Outcome B (cont.)** Learning Outcome C: Using characters to manipulate data in programs.
Tuesday	25 Jul	
Wednesday	26 Jul	
Thursday	27 Jul	
Friday	28 Jul	
Monday	31 Jul	**Class Test 1**
Tuesday	1 Aug	Learning Outcome D.1. : Using strings to manipulate data in programs.
Wednesday	2 Aug	
Thursday	3 Aug	
Friday	4 Aug	
Monday	7 Aug	**Class Test 2**
Tuesday	8 Aug	Learning Outcome D.1. : Using strings to manipulate data in programs.
Wednesday	9 Aug	**No Class - Woman's Day**
Thursday	10 Aug	**Feedback on CT1** Learning Outcome D.1. : Using strings to manipulate data in programs. **More Exercises on learning Outcome D (cont.)**
Friday	11 Aug	
Monday	14 Aug	**Class Test 3**
Tuesday	15 Aug	**Feedback on CT2**
16 – 26 Aug TEST WEEK 1		
Monday	28 Aug	**CT 3 and Semester Test 1/Formative Assessment Feedback** Learning Outcome E.1- Learning Outcome E.1.11. : Implementing one-dimensional (1-D) arrays to store data in programs (cont.).
Tuesday	29 Aug	
Wednesday	30 Aug	

Thursday	31 Aug	**Exercises on learning Outcome E.1.1- E.1.11**
Friday	1 Sep	
Monday	4 Sep	Learning Outcome E.1.12- Learning Outcome E.1.18. : Implementing one-dimensional (1-D) arrays to store data in programs (cont.) **Exercises on learning Outcome E.1.11- E.1.14 (cont.)**
Tuesday	5 Sep	
Wednesday	6 Sep	
Thursday	7 Sep	
Friday	8 Sep	
Monday	11 Sep	**Class Test 4** **Assessment Scope Learning Outcome F.1- F.1.14** Learning Outcome E.1.18- Learning Outcome E.1.23 : Implementing one-dimensional (1-D) arrays to store data in programs (cont.)
Tuesday	12 Sep	
Wednesday	13 Sep	
Thursday	14 Sep	
Friday	15 Sep	
Monday	18 Sep	**Class Test 4 Feedback**
	18 – 30 Sep	**TEST WEEK 2**
	2- 7 Oct	**RECESS**
Monday	9 Oct	Learning Outcome F.1.11- Learning Outcome F.1.19 : Creating user-defined classes in programs to promote software reuse.
Tuesday	10 Oct	
Wednesday	11 Oct	
Thursday	12 Oct	
Friday	13 Oct	
Monday	16 Oct	**Class Test 5** Learning Outcome F.1.11- Learning Outcome F.1.19 : Creating user-defined classes in programs to promote software reuse (cont.). **More Exercises on learning Outcome F (cont.)**
Tuesday	17 Oct	
Wednesday	18 Oct	
Thursday	19 Oct	
Friday	20 Oct	
Monday	23 Oct	**Class Test Feedback** Learning Outcome F.1.11- Learning Outcome F.1.19 : Creating user-defined classes in programs to promote software reuse (cont.). **More Exercises on learning Outcome F (cont.)** **Revision**
Tuesday	24 Oct	
Wednesday	25 Oct	
Thursday	26 Oct	
Friday	27 Oct	
Monday	30 Nov	**Predicate Day**
Tuesday	1 Nov	**Revision**
Wednesday	2 Nov	
Thursday	3 Nov	
Friday	4 Nov	
	6- 24 Nov	**Examinations**
	27 Nov- 8 Dec	**Sick Test and mark finalization**
		TUT RECESS

SECTION C – PRESCRIBED AND RECOMMENDED RESOURCES

15. Prescribed Textbook



Java How to Program – Late Objects Version

By Paul Deitel and Harvey Deitel.

Published by Pearson Education Limited

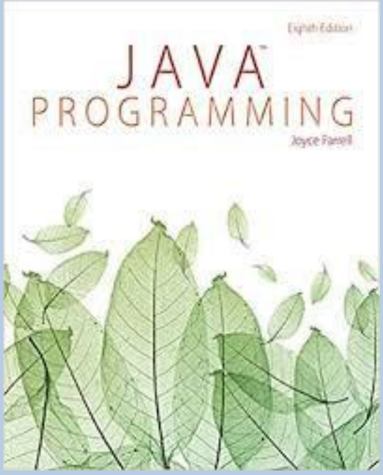
ISBN-10: 978-1-292-01936-9

This book is essential for the successful completion of this course. You are strongly advised to acquire the resource.

16. Additional Module Resource Information

- Student may be given handouts from other relevant books.
- Computers

17. Recommended Additional Resources



Java Programming

By Joyce Farrell.

Eight Edition

Published by Cengage Learning
ISBN: 978-1-285-85691-9

18. Recommended Electronic resources

Learning outcome, or assessment criteria that this resource relates to	Title of electronic resource	URL
N/A		

19. Supplied Resources

There are no resources supplied for this module.

20. E-Learning platform

This module will make use of the new **MyTutor (Bright Space)** (https://mytutordsl.tut.ac.za)as its formal e-learning platform. Last minute, important notifications, like changes in lecture venue, time, deadlines for assignments, etc. will be placed on this platform. Electronic resources will also be distributed via this platform. You must make use of your TUT4Life email for communication purposes in this subject. Electronic Campus (also known as **EC**) will be used for assessment purposes only if needed.

SECTION D – MODULE DESCRIPTION

21. Alignment of this module with the relevant exit level outcomes / graduate attributes

The students should be able to do the following after the completion of the module:

- Scope of knowledge, in respect of which a learner is able to demonstrate: detailed knowledge of the main areas of one or more fields, disciplines or practices, including an understanding of and an ability to apply the key terms, concepts, facts, principles, rules and theories of that field, discipline or practice to unfamiliar but relevant contexts; and knowledge of an area or areas of specialisation and how that knowledge relates to other fields, disciplines or practices
- Knowledge literacy, in respect of which a learner is able to demonstrate a understanding of different forms of knowledge, schools of thought and forms of explanation within an area of study, operation or practice, and an awareness of knowledge production processes
- Method and procedure, in respect of which a learner is able to demonstrate an ability to evaluate, select and apply appropriate methods, procedures or techniques in processes of investigation or application within a defined context
- Problem solving, in respect of which a learner is able to demonstrate an ability to identify, analyse and solve problems in unfamiliar contexts, gathering evidence and applying solutions based on evidence and procedures appropriate to the field, discipline or practice
- Ethics and professional practice, in respect of which a learner is able to demonstrate an understanding of the ethical implications of decisions and actions, within an organisational or professional context, based on an awareness of the complexity of ethical dilemmas f. Accessing, processing and managing information, in respect of which a learner is able to demonstrate an ability to evaluate different sources of information, to select information appropriate to the task, and to apply well-developed processes of analysis, synthesis and evaluation to that information
- Producing and communicating information, in respect of which a learner is able to demonstrate an ability to present and communicate complex information reliably and coherently using appropriate academic and professional or occupational conventions, formats and technologies for a given context
- Context and systems, in respect of which a learner is able to demonstrate an ability to make decisions and act appropriately in familiar and new contexts, demonstrating an understanding of the relationships between systems, and of how actions, ideas or developments in one system impact on other systems

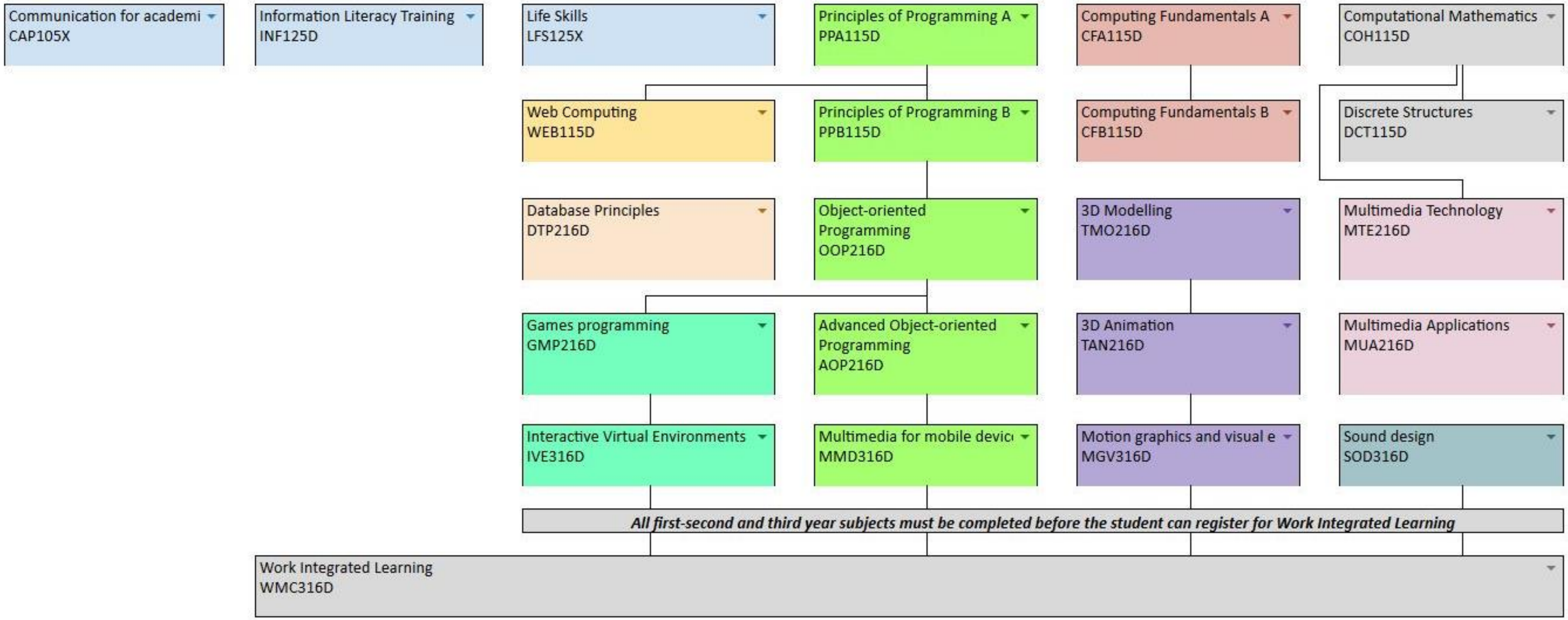
Management of learning, in respect of which a learner is able to demonstrate an ability to evaluate performance against given criteria, and accurately identify and address his or her task-specific learning needs in a given context, and to provide support to the learning needs of others where appropriate

- Accountability, in respect of which a learner is able to demonstrate an ability to work effectively in a team or group, and to take responsibility for his or her decisions and actions and the decisions and actions of others within well-defined contexts, including the responsibility for the use of resources where appropriate.

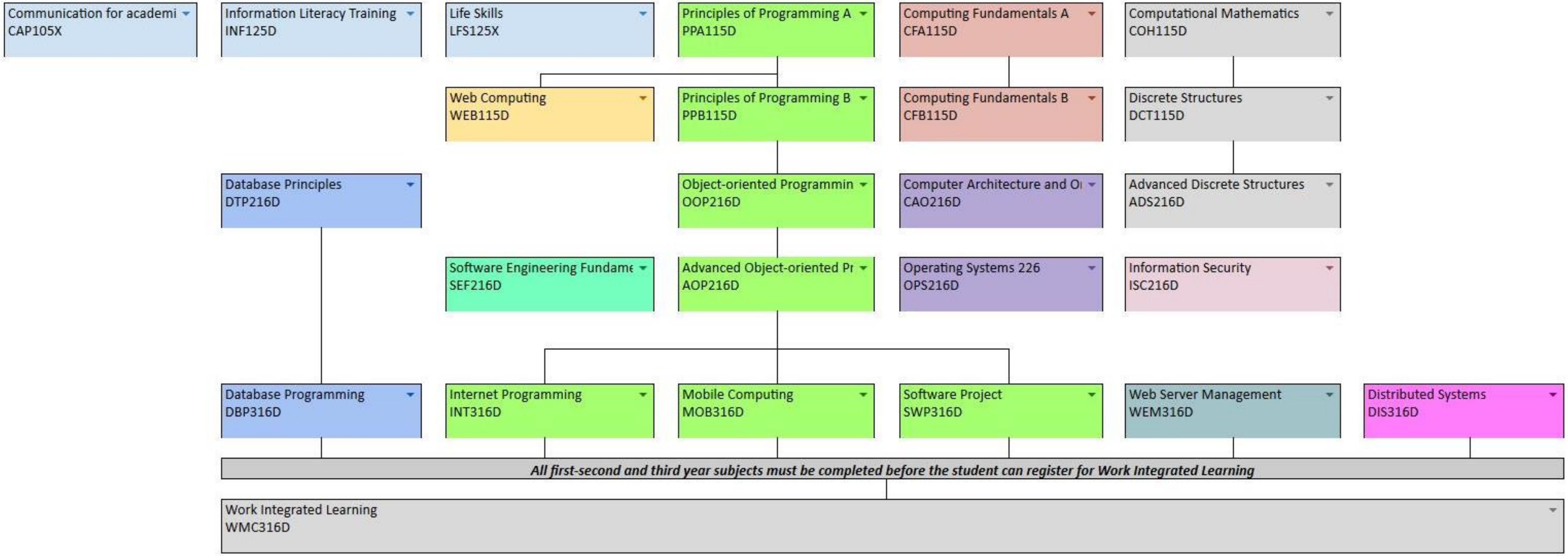
22. Articulation with other modules in the programme

The diagrams below show the articulation form one module to the next. If the module is connected with a line to a module above it, it indicates that the module above is a prerequisite for this module.

22.1. Diploma in Multimedia Computing

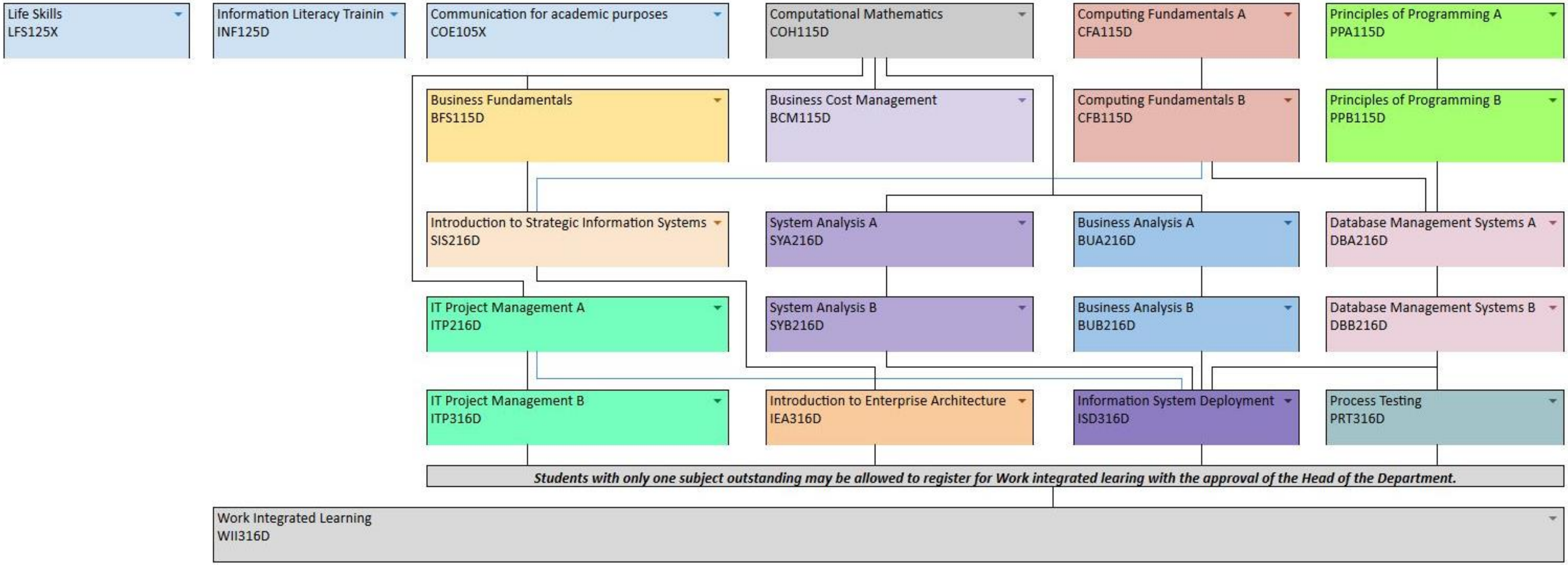

Communication for academi
CAP105X
Information Literacy Training
INF125D
Life Skills
LFS125X
Principles of Programming A
PPA115D
Computing Fundamentals A
CFA115D
Computational Mathematics
COH115D
Web Computing
WEB115D
Principles of Programming B
PPB115D
Computing Fundamentals B
CFB115D
Discrete Structures
DCT115D
Database Principles
DTP216D
Object-oriented Programming
OOP216D
3D Modelling
TMO216D
Multimedia Technology
MTE216D
Games programming
GMP216D
Advanced Object-oriented Programming
AOP216D
3D Animation
TAN216D
Multimedia Applications
MUA216D
Interactive Virtual Environments
IVE316D
Multimedia for mobile devic
MMD316D
Motion graphics and visual e
MGV316D
Sound design
SOD316D
All first-second and third year subjects must be completed before the student can register for Work Integrated Learning
Work Integrated Learning
WMC316D

22.2. Diploma in Computer Science


Communication for academi CAP105X
Information Literacy Training INF125D
Life Skills LFS125X
Principles of Programming A PPA115D
Computing Fundamentals A CFA115D
Computational Mathematics COH115D
Web Computing WEB115D
Principles of Programming B PPB115D
Computing Fundamentals B CFB115D
Discrete Structures DCT115D
Database Principles DTP216D
Object-oriented Programmin OOP216D
Computer Architecture and O CAO216D
Advanced Discrete Structures ADS216D
Software Engineering Fundame SEF216D
Advanced Object-oriented Pr AOP216D
Operating Systems 226 OPS216D
Information Security ISC216D
Database Programming DBP316D
Internet Programming INT316D
Mobile Computing MOB316D
Software Project SWP316D
Web Server Management WEM316D
Distributed Systems DIS316D
Work Integrated Learning WMC316D

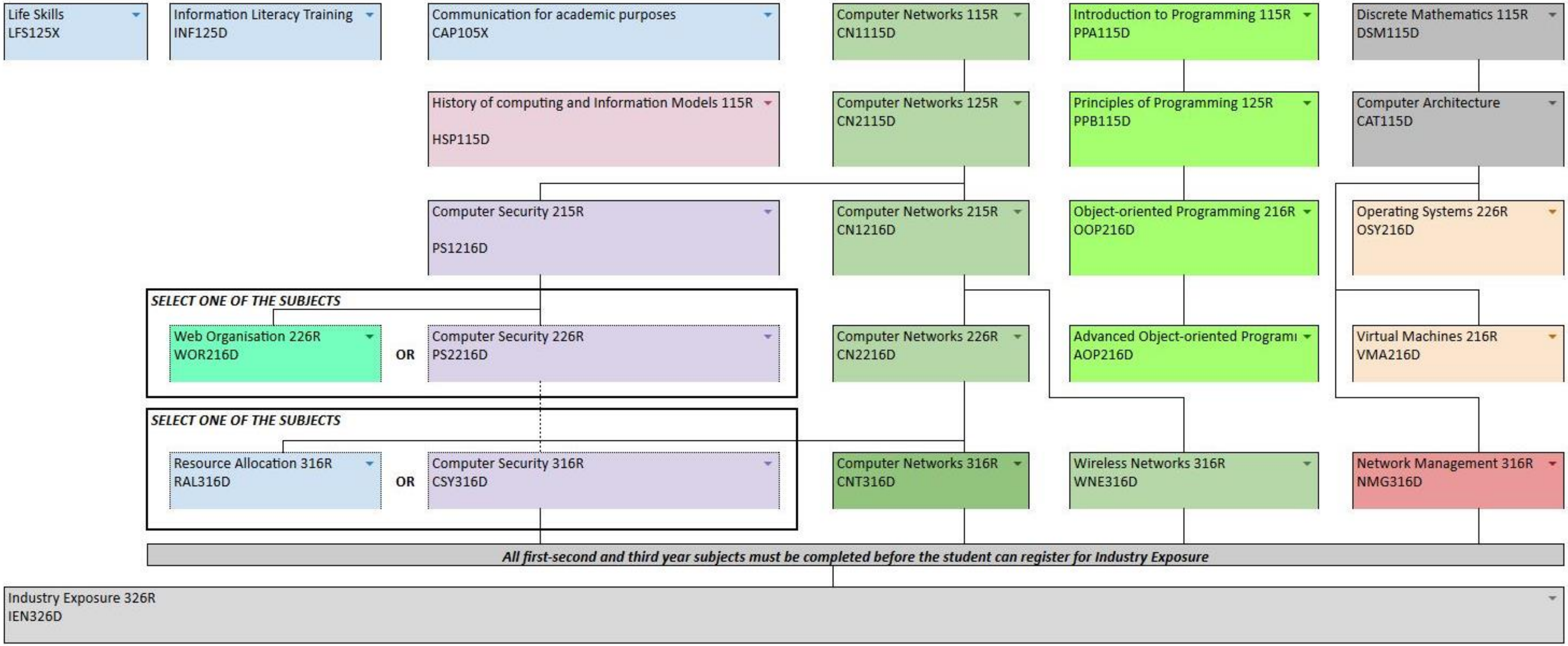
All first-second and third year subjects must be completed before the student can register for Work Integrated Learning

22.3. Diploma in Informatics



Students with only one subject outstanding may be allowed to register for Work integrated learing with the approval of the Head of the Department.

22.4. Diploma in Information Technology



Life Skills LFS125X

Information Literacy Training INF125D

Communication for academic purposes CAP105X

Computer Networks 115R CN1115D

Introduction to Programming 115R PPA115D

Discrete Mathematics 115R DSM115D

History of computing and Information Models 115R HSP115D

Computer Networks 125R CN2115D

Principles of Programming 125R PPB115D

Computer Architecture CAT115D

Computer Security 215R PS1216D

Computer Networks 215R CN1216D

Object-oriented Programming 216R OOP216D

Operating Systems 226R OSY216D

SELECT ONE OF THE SUBJECTS

Web Organisation 226R WOR216D **OR** Computer Security 226R PS2216D

Computer Networks 226R CN2216D

Advanced Object-oriented Programı AOP216D

Virtual Machines 216R VMA216D

SELECT ONE OF THE SUBJECTS

Resource Allocation 316R RAL316D **OR** Computer Security 316R CSY316D

Computer Networks 316R CNT316D

Wireless Networks 316R WNE316D

Network Management 316R NMG316D

All first-second and third year subjects must be completed before the student can register for Industry Exposure

Industry Exposure 326R IEN326D

23. Module Outline

This module will comprise of the following **learning goals**:

A. Revision on

   A.1.1 Using selection control structure in programs.

   A.1.2 Using iteration control structures in programs.

B. Creating modularised interactive programs.

C. Using characters to manipulate data in programs.

D. Using strings to manipulate data in programs.

E. Implementing one-dimensional arrays to store data in programs.

F. Implementing two-dimensional arrays to store data in programs.

G. Creating user-defined classes in programs to promote software reuse.

The learning outcomes, and assessment criteria for how you can measure if you have reached a full understanding of these learning goals are defined in the next section.

24. Learning outcomes, assessment criteria, teaching activities, and Assessment method

Each **learning goal** has a set of **learning outcomes** that will be assessed using **assessment criteria**. The tables below show the learning outcomes, assessment criteria, teaching activities, and assessment methods.

Learning outcome: A.1. Implementing selection control structures in programs. *(Reference Book: Notes will be provided.)*

Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
A.1.1 - By implementing the “if” selection control structure, the ability of a program to make a decision is created according to the Java language syntax rules. A.1.2 - By implementing the “if…else” selection control structure, the ability of a program to make a decision is created according to the Java language syntax rules. A.1.3 - The importance of the “{}” in an if selection control structure is discussed in terms of program modifiability. A.1.4 - The nested if selection structure is used to make sub decisions within decisions. A.1.5 - A safely modifiable program is created to avoid implementing the “if…else if” selection control structure within the program. A.1.6 - Two values are compared using the “>, <, >=, <=, ==, !=” relational operators. A.1.7 - In an if statement, the “&&, ||,!” Boolean/logical operators are applied to combine the results of multiple relational conditions. A.1.8 - Optimised conditional statements are written using operator short-circuiting.	Lecturer controlled: Peer controlled: Student controlled:	**Formative:** - Class discussions **Summative:** - Class tests (computer based) - Group Assignments - Semester tests - Examination.

A.1.9 - By implementing the “switch” selection control structure, the ability of the program to make decisions is created according to the Java language syntax rules. A.1.10 - The break keyword is used inside a switch selection control structure to terminate the switch statement block. A.1.11 - The nested “switch” selection control structure is used to make sub decisions within decisions. A.1.12 - By implementing all of the above mentioned selection control structures, the ability of the program to make decisions is created according to Java language syntax rules. A.1.13 - IPO charts and flowcharts are used to create programs that implement selection control structures.		

Learning outcome: A.2. Implementing iteration control structures in programs. *(Reference Book: Chapter 3 and Chapter 4 and class notes.)*

Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
A.2.1 - A differentiation is made between a count-controlled, sentinel-controlled and infinite loop. A.2.2 - A while repetition control structure is used to repeat a block of statements based on program design requirements. A.2.3 - A do…while repetition control structure is used to repeat a block of statements based on program design requirements. A.2.4 - A for repetition control structure is used to repeat a block of statements based on program design requirements. A.2.5 - Loops are nested to display grid information. A.2.6 - The concept of accumulation is demonstrated by increasing a value within repetition control structures.	Lecturer controlled: Peer controlled: Student controlled:	**Formative:** - Class discussions **Summative:** - Class tests (computer based) - Group Assignments - Semester tests - Examination.

A.2.7 - By implementing all of the above mentioned iteration control structures, the ability of the program to repeat blocks of statements is created according to Java language syntax rules .		

Learning outcome: B.1. Creating modularised interactive programs. *(Reference Book: Chapter 5 and class notes)*		
Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
B.1.1 - A definition of modularisation is given. B.1.2 - A definition of a method is given. B.1.3 - An explanation of the static keyword is given. B.1.4 - A description of the method header of the main method is given. B.1.5 - A differentiation between static and non-static methods is given. B.1.6 - A UML class diagram that explains the representation of static members and methods is created. B.1.7 - A static method that requires parameters and returns data is created in the frontend class. B.1.8 - A static method that requires parameters but returns no data is created in the frontend class. B.1.9 - A static method that requires no parameters but returns data is created in the frontend class. B.1.10 - A static method, which requires arguments and returns data, is invoked in the main method of the frontend class. B.1.11 - A static method, which requires arguments but returns no data, is invoked in the main method of the frontend class. B.1.12 - A static method, which requires no arguments but returns data, is invoked in the main method of the frontend class.	rer controlled: controlled: ent controlled:	**ative:** Class discussions **native:** Class tests (computer based) Group Assignments Semester tests Examination.

Learning outcome: C.1. Using characters to manipulate data in programs. *(Reference Book: Chapter 14.2 and 14.5 and class notes)*		
Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
C.1.1 - The **ASCII table** is explained. C.1.2 - The **ASCII** value of a character is printed on the screen. C.1.3 - The **isDigit()** method of the Character class is used to determine if a character is a digit. C.1.4 - The **isLetter()** method of the Character class is used to determine if a character is a letter. C.1.5 - The **isLetterOrDigit()** method of the Character class is used to determine if a character is a letter or digit. C.1.6 - The **isLowerCase()** method of the **Character** class is used to determine if a character is lower case. C.1.7 - The **isUpperCase()** method of the **Character** class is used to determine if a character is upper case. C.1.8 - The **isWhitespace()** method of the **Character** class is used to determine if a character is a whitespace. C.1.9 - The **toLowerCase()** method of the **Character** class is used to convert a character to lower case. C.1.10 - The **toUpperCase()** method of the **Character** class is used to convert a character to upper case. C.1.11 - The **toString()** method of the **Character** class is used to convert a character to a string.	Lecturer controlled: Peer controlled: Student controlled:	**Formative:** - Class discussions **Summative:** - Class tests (computer based) - Group Assignments - Semester tests - Examination.

Learning outcome: D.1. Using strings to manipulate data in programs. *(Reference Book: Chapter 14.1 – 14.4 and class notes)*

Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
D.1.1 - A determination is made whether a **String** is an object or a primitive data type. D.1.2 - A **String** is declared using the shorthand method according to the Java language syntax rules. D.1.3 - A determination is made of how characters in a String are indexed by using the **charAt**() method of the String class to demonstrate it. D.1.4 - A new String is created by converting the characters of an existing String to upper case using the **toUpperCase**() method of the String class. D.1.5 - A new String is created by converting the characters of an existing String to lower case using the **toLowerCase**() method of the String class. D.1.6 - The **length()** method of the String class is used to determine the length of a String D.1.7 - A new String is created by using the **substring**() method of the String class to extract a portion of an existing String. D.1.8 - The index of the first occurrence of a specified character/s within a String is determined by using the **indexOf**() method of the String class. D.1.9 - The index of the last occurrence of a specified character/s within a String is determined by using the **lastIndexOf**() method of the String class. D.1.10 - A character/s is replaced with another character within an existing String by using the **replace**() method of the String class. D.1.11 - A String is compared to another String for equality using the **equals**() method of the String class. D.1.12 - A String is compared to another String for equality using the **equalsIgnoreCase**() method of the String class. D.1.13 -A String is compared to another String for equality using the **compareToIgnoreCase()** method of the String class.	Lecturer controlled: Peer controlled: Student controlled:	**Formative:** - Class discussions **Summative:** - Class tests (computer based) - Group Assignments - Semester tests - Examination.

D.1.14 - A new String is created by concatenating two strings using the **concat**() method of the String class. D.1.15 - Tokens are extracted from a string using the **split()** method of the String class. D.1.16 - A String is converted into an integer using the **Integer.parseInt()** method of the Integer wrapper class. D.1.17 - A String is converted into a double using the **Double.parseDouble()** method of the Double wrapper class. D.1.18 - A String is converted into a float using the **Float.parseFloat()** method of the Float wrapper class. D.1.19 - An integer is converted into a String using the **String.valueOf()** method of the String wrapper class. D.1.20 - A float is converted into a String using the **String.valueOf()** method of the String wrapper class. D.1.21 - A double is converted into a String using the **String.valueOf()** method of the String wrapper class.		

Learning outcome: E.1. Implementing one-dimensional (1-D) arrays to store data in programs. *(Reference Book: Chapter 6)*		
Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
E.1.1 - A one-dimensional array is defined. E.1.2 - The properties of a one-dimensional array are explained. E.1.3 - A one-dimensional array of primitives is declared. E.1.4 - A one-dimensional array of Strings is declared. E.1.5 - A one-dimensional array of primitives is **declared** and **initialised** using one instruction.	Lecturer controlled: Peer controlled: Student controlled:	**Formative:** - Class discussions **Summative:** - Class tests (computer based) - Group Assignments - Semester tests

E.1.6 - A one-dimensional array of Strings is declared and initialised using one instruction. E.1.7 - A one-dimensional array of primitives is populated with data after it was declared using the index notation. E.1.8 - A one-dimensional array of Strings is populated with data after it was declared using the index notation. E.1.9 - The size of a one-dimensional array is dynamically determined. E.1.10 - The elements of a one-dimensional array are accessed and processed using the index notation. E.1.11 - The data **accumulation** concept is demonstrated using a one-dimensional array. E.1.12 - The parallel arrays concept is explained. E.1.13 - Parallel arrays are created to store related data. E.1.14 - Parallel arrays are passed to methods for processing. E.1.15 - The linear search technique is explained. E.1.16 - The linear sort technique is used to search a one-dimensional array. E.1.17 - The bubble sort technique is explained. E.1.18 - The bubble sort technique is used to sort the elements of one-dimensional array. E.1.19 - The pass-by-value concept is explained. E.1.20 - The pass-by-reference concept is explained. E.1.21 - A determination is made of what happens to the contents of an array after it is passed to a method for manipulation. E.1.22 - A one-dimensional array is passed to methods for processing. E.1.23 - An array created within a method is returned from the method.		- Examination.

Learning outcome: F.1. Creating user-defined classes in programs to promote software reuse. *(Reference Book: Chapter 7 and class notes)*		
Assessment Criteria *(noun + verb + measurable standard)*	**Teaching activities**	**Assessment method**
F.1.1 - The class concept is defined. F.1.2 - A differentiation is made between a class and an object. F.1.3 - A determination is made of what a class structure consists of according to the Java language syntax rules. F.1.4 - A Java class is created using the Java language syntax rules. F.1.5 - A private **data member** is declared based on the Java naming convention for class members. F.1.6 - A public **data member** is declared based on the Java naming convention for class members F.1.7 - Explain the representation of data members, methods and constructors in a UML class diagram. F.1.8 - A **method** is named according to the Java naming convention. F.1.9 - A **mutator (set)** method is written to change a value of a private data member. F.1.10 - An **accessor (get)** method is written to extract the value of a private data member. F.1.11 - A method is written to receive values through input parameters and return a new calculated value. F.1.12 - A method that receives more than one input argument is written. F.1.13 - A method that receives one or more object/s as an input argument/s is written. F.1.14 - The purpose of a **constructor** is determined according to OOP design. F.1.15 - A **default constructor** is written to initialize private data members to default value. F.1.16 - A **constructor** that receives one or more input arguments is written to initialize private data members.	Lecturer controlled: Peer controlled: Student controlled:	**Formative:** - Class discussions **Summative:** - Class tests (computer based) - Group Assignments - Semester tests - Examination.

F.1.17 - The concept of reference data types is discussed. F.1.18 - A class is instantiated in the main method of a frontend class. F.1.19 - The members of an object are accessed in the main method of the frontend class.		

As presented in the tables above, each learning outcome will be assessed with an assessment method. Below is a table that describes what these assessment methods entail.

Assessment Method	Example/description	Possible uses/assessment tasks	Where to use
Alternative response questions	True/False; Yes/No questions; multiple choice	• Recall of information • Ability to discriminate	• Formative • Summative
Assertion/reason questions	Consists of an assertion and supporting explanation. The learner has to decide whether the assertion and explanation are true, and if true, whether the explanation is a valid reason for the assertion. Sometimes the learner is asked to select his/her answer from a list of possibilities, e.g. True; True + Valid; True + Invalid.	• Ability to weigh up options and to discriminate	• Formative • Summative • RPL
Aural/oral tests	These are mainly used to generate evidence on learners' ability to listen, interpret, communicate ideas and sustain a conversation in the language of assessment.	• Interpretation of ideas • Expression of ideas	• Formative • Summative • RPL
Completion questions/short answer questions	Learners are presented with a question with a pre-determined answer consisting of a few words, or may be given a statement where key words are omitted. They are then required to complete the statement by filling in the word(s). Such questions may also involve the use of numbers, diagrams and graphs.	• Recall of factual information • Test understanding and application of knowledge, e.g. in mathematical concepts	• Formative • Summative • RPL
Examinations / tests	These usually consist of a range of questions. Learners are required to respond to questions within a specified time.	• Recall of information • Cognitive skills such as problem solving or analyses	• Formative • Summative • RPL

Extended response questions	These are usually in a written form. There are few restrictions on the content and form of the response. Continuous prose is normally required, but there may be limits on the length and/or time allocated.	• Open-ended debates or other responses • Arguments • Reports	• Formative • Summative • RPL
Grid questions /matching questions	Grid questions and matching questions are variants of each other. The learner is presented with two lists – a set of statements and a set of responses. The learner is required to indicate which response from the second list corresponds or matches each statement in the first list. Grid questions are presented in grid format. They differ from the other selected-response assessment instruments in that each question may have more than one correct response and each response may be used more than once.	• Recall of information • Application of knowledge	• Formative • Summative • RPL
Multiple choice questions	Multiple choice questions consist of an incomplete statement or a question, followed by plausible alternative responses from which the learner has to select the correct one. Outcomes involving higher order analytical skills are probably more validly assessed by means of free-response assessment instruments such as extended response questions, but multiple-choice questions can be useful if carefully constructed.	• Recall of information • Check understanding; analyses	• Formative • Summative • RPL
Oral questions / restricted response questions	The form and content of the response is limited by the way in which the question is asked. These questions do not have pre-determined correct answers (as in short answer questions, etc.) and the assessor has to exercise his/her professional judgment when interpreting learner's responses.	• Allows for self-expression when questions are oral Supports observation of tasks where underpinning knowledge and • understanding are tested	• Formative • Summative (small groups only) • RPL
Personal interviews	A personal interview is probably the oldest and best-known means of eliciting information directly from learners. It combines two assessment methods, namely observation and questioning. An interview is a dialogue between the assessor and the learner, creating opportunities for learner questions.	• A range of applications using different forms of questions, particularly • Open-ended questions • Guidance and support to the learner	• Formative • Summative (small groups only) • RPL
Questionnaires	A questionnaire is a structured written interview consisting of a set of questions relating to particular areas of performance. Unlike a personal interview, it is administered and judged under standard conditions.	• Assessment of outcomes particularly concerned with attitudes, feelings, interests and experiences	• Formative • Summative • RPL

Structured questions	A structured question consists of a stem (which describes a situation), followed by a series of related questions. The stem can be text, a diagram, a picture, a video, etc.	• Recall of information • Application of knowledge and understanding • Analyses, Debates, Arguments	• Formative • Summative • RPL
Assignments	A problem-solving exercise with clear guidelines and a specified length. More structured and less open-ended than projects, but they do not necessarily involve strict adherence to a prescribed procedure, and they are not concerned exclusively with manual skills.	• Problem-solving around a particular topic	• Formative • Summative • RPL
Case studies	A description of an event concerning a real-life or simulated situation, usually in the form of a paragraph or text, a video, a picture or a role-play exercise. This is followed by a series of instructions to elicit responses from learners. Individuals or small groups may undertake case studies.	• Analyses of situations • Drawing conclusions • Reports on possible courses of action	• Formative • Summative • RPL
Logbooks	A useful means of assessing learner's progress and achievements. It should have clear instructions for use and give guidance on how essential information is to be recorded.	• In a workplace – monitor and check activities; record processes; record of achievements	• Formative • Summative • RPL • (Learner has to be in workplace)
Practical exercises / demonstrations	An activity that allows learners to demonstrate manual and/or behavioural skills. The assessment may be based on the end-result of the activity (the product), or the carrying-out of the activity (the process), or a combination of both	• Demonstration of skill	• Formative • Summative • RPL (Not always practical – logistics)
Portfolios	A collection of different types of evidence relating to the work being assessed. It can include a variety of work samples. Portfolios are suitable for long-term activities. It is important that the evidence in the portfolio meet the requirements of sufficiency and currency. The learner and assessor usually plan the portfolio jointly as sources of evidence may vary. The learner is then responsible for the collection of evidence and the compilation of the portfolio.	• Recognition of prior learning and experience • Assessment of long-term activities related to each other • Assessment where direct observation may be difficult	• Summative • RPL • (Not recommended for formative)
Projects	A project is any exercise or investigation in which the time constraints are more relaxed. Projects are: • Practical • Comprehensive and open-ended • Tackled without close supervision, but with assessor guidance and support	• Comprehensive range of skills can be assessed • Integration of activities within and across unit standards or different parts of a qualification	• Summative • RPL • (Not recommended for formative)

	Projects can involve individuals or a group of learners. The choice of the project is directed by the assessor, usually by providing the learner with a topic or brief for the investigation		
Role-plays	Learners are presented with a situation, often a problem or an incident, to which they have to respond by assuming a particular role. The enactment may be unrehearsed, or the learner may be briefed in the particular role to be played. Such assessments are open-ended and are person centred.	• Assessment of a wide range of behavioural and interpersonal skills	• Formative • Summative • RPL
Reflective journal	A reflective journal gives learners the opportunity to critically reflect on their own learning, to express their thoughts and experiences and to present this in an acceptable way. Even though it is a form of self-assessment, it can be submitted for assessment.	• It gives the assessor a unique opportunity to follow the thought-processes of a learner and to monitor the way a learner thinks and grows • Critical evaluation of progress by the learner	• Formative
Self-assessment	A checklist, questionnaire completed by a learner, notes jotted down, or other forms of structured self-assessment undertaken after an action, demonstration, oral examination, etc.	• Critical evaluation of progress by learner	• Formative
Peer assessment	Assessment by the learner's peers, usually in the form of a checklist.	• Assessment of paired or group activities • Assessment of teamwork	• Mainly formative • Summative if team-, group work part of outcomes

SECTION E – ASSESSMENT

25. Assessment Schedule

The mode of assessment for Programming Principles is Examination. Students will write semester tests, class tests, web test, and class exercises and submit these assessments which will each contribute to a predicate. The purpose of assessment is to determine whether you have achieved the learning outcomes. The various assessment methods therefore will focus on criteria that will enable the lecturer(s) to determine whether you have achieved the learning outcomes. Based on the predicate, a student can either qualify for examination entrance or not (40%).

Your predicate will consist of a combination of marks from the following assessment types:

25.1 Class Tests / Web Tests

Five class test will be written as indicated in the schedule. There will also be a number of surprise class tests that might be written any time during class time. Students must ensure that they attend all classes to participate in these assessment opportunities. Failure to write a class will result in a student getting a zero for the test.

There is **No sick test** provision for class tests. It might be possible that a smaller number of class test are written if some unforeseen event arises. **If there any queries regarding the tests, these queries must be submitted to the lecturer on the day when the test is discussed.**

25.2 Formal tests

You will be evaluated on two invigilated formal tests, which will be written during the two test weeks and each will contribute 30% towards your predicate.

Students, who did not write one of the formal semester tests due to a reason other than sickness or extraordinary circumstances, he/she will ONLY be allowed to do a sick test if the necessary acceptable documents of proof are submitted to the lecturer within one week after the semester test. The sick test will be written on all the work done shortly before predicate day. **If there any queries regarding the semester tests, these queries must be submitted to the lecturer on the day when the test is being discussed.**

Each formal test will count a certain percentage towards the student's predicate mark as shown in Section 27**.**

25.3 Assignments

There are no assignments in this module.

25.4 Sick Test

There will be no sick test for a **class test** missed.

If a formal test was not written due to illness, a valid doctor's certificate needs to be presented within 5 days after the assessment was conducted. A supplementary test may be allowed under exceptional circumstances if proper proof was provided to the lecturer within 5 days after the assessment was conducted.

The **Sick test** will be written at the end of the second semester (before the predicate mark is calculated). Please note that only one sick test will be allowed and that the mark for this test will replace the mark of the test that was missed.

The **supplementary** exam will be written at the end of the semester after the main exam as is indicated in section 14. Please note that the mark for this supplementary paper will replace the mark of the main exam paper.

The sick test will be scheduled as per work schedule of this guide.

26. Assessment Schedule and Scope

Date	Assessment Task	Learning Outcomes and assessment criteria to be assessed
31 Jul	**Class Test /Web Test 1**	Learning Outcome A (Control Structures)
7 Aug	**Class Test /Web Test 2**	Learning Outcome B (Modularization)
14 Aug	**Class Test /Web Test 3**	Learning Outcome C – Learning Outcome D (Characters and Strings)
11 Sep	**Class Test /Web Test 4**	Learning Outcome E (One dimensional)
16 Oct	**Class Test /Web Test 5**	Learning Outcome F (Use defined classes)
Check Time table	**Formative assessment 1**	To be announced
Check Time table	**Formative assessment 2**	All the work
Check Time table	**Summative assessment (Exam)**	All the work

If you have queries about your mark, you must immediately consult your lecturer after test has been written.

The predicate mark and the examination mark each contribute 50% towards the final mark as shown in the table below.

Date	Assessment Task	Weight towards year mark	Learning outcomes and assessment criteria to be assessed
	Predicate	50%	All Learning Outcome
(Please see faculty time table)	Exams	50%	All Learning Outcome
	Final Mark	**100%**	

27. Composition of Final Mark

The weights of the various assessments may change during the semester due to unforeseen circumstances. These changes will be discussed with the students if and when they should arise.

	Assessment	*Weight*	*Final Mark*
SW	Class Test 1	**40%**	**50%**
	Class Test 2		
	Class Test 3		
	Class Test 4		
	Class Test 5		
	Surprise Class Tests/Web Tests		
FA	Formative Assessment 1	**30%**	
	Formative Assessment 2	**30%**	
Exam/Sup	**Exam**		**50%**
	Total		**100%**

PLEASE TAKE NOTE:

All tests marked will be returned and discussed during class. If you do not attend the class, your test will not be available for discussion later.

Predicate marks are put on the faculty notice boards. **If you have queries about your mark, you must immediately consult your course lecturer (contact details are given above), before predicate day.**

28. Assessment Administration

Test administration:

1. Students will write the test.
2. The Lecturer/s will mark the test.
3. The Lecturer will hand back the marked scripts to each student and discuss common problems encountered. At the end of this session the marked scripts need to submitted back to the lecturer.
4. If there are any queries regarding the test, **these queries must be submitted to the lecturer within 5 days** after the marked scripts were discussed. **NO marks changes will be made after 5 days.**

29. Promotional requirements

To qualify to write the examination, a predicate mark of 40% must be obtained. To pass the module, a student must obtain a final mark of at least 50% as final mark, with a sub minimum of 40% for the examination. If a student fails the subject, but obtains a final mark of at least 45%, he/she qualifies for a supplementary examination. Students with a final mark of 50% or more, yet who failed to obtain the sub minimum of 40% in the examination, will also qualify for a supplementary examination.

It is the responsibility of the student to consult the official examination notice boards/electronic system to find out whether he/she qualifies for a supplementary examination.

30. Absence from assessment opportunities or late submission of assessments

If a formal test was not written/submitted due to illness, a valid doctor’s certificate needs to be presented within 5 days after the assessment was conducted. A supplementary test/assignment may be allowed under exceptional circumstances if proper proof was provided to the lecturer within 5 days after the assessment was conducted. Please not that at the end of each assessment, the method submission will be communicated. For some assessments you might be required to submit your assessment via a web interface. Failure to do so correctly will result in a zero mark. No supplementary test/assignment will be allowed if you fail to submit your assessment according to the submission instructions stated in the assessment question.

There is no supplement assessment that will be written at the end of the semester.

If a student is left with one module remaining, he/she may apply for an exit exam. The exit exam will be written at the end of the semester. The dates on when this paper is to be written will be communicated by the exams department. Please note that a student may only write the exit exam. Please note that the mark for this exam paper will replace the mark of the exam paper that was failed.

31. Mode of delivery

The mode of delivery for this module is contact. This module will be presented in 4 periods per week during the day.

From 5PM each day, the computer laboratories will be made available for students to practice and do homework. Student assistants are made available in these laboratories to assist students with homework related problems. These assistants are not just there to open and close the labs, please make use of their assistance.

32. Quality assurance

The quality of the study guide is maintained by the Quality Assurance committee of Tshwane University of Technology.

Students evaluate the study guide using the lecturer assessment questionnaire that is completed twice a semester.

The quality of the question papers and memorandums is maintained by an external moderator and internal moderator with qualifications higher than the year the module is being presented in.

The quality of the marking is overseen by the external moderator that ensures the marking of the scripts is fair. A moderator report is completed for each major assessment.

33. Industry related learning

Where possible guest speakers will come and address the class on advanced programming issues.

34. Plagiarism

The following is an extract from TUT's plagiarism policy. Reference: RIPPOL067.

All students have a moral obligation and responsibility to maintain the following academic integrity principles in the production and presentation of academic outputs, regardless of the presentation format and/or work type:

Each student should only submit his/her own original academic work, except when formal group work was required in the production of the academic output;

Each student should accurately indicate in all academic outputs when information is used that was produced by another scholar by referencing it in accordance with a recognised referencing convention system;

No student should use, present or submit someone else's electronic works, multimedia products or artistic works as if it is his/her own;

Each student should accurately indicate the download/access date and the uniform resource locator (URL) of the internet web page when information is used from a website, web page or other electronic source;

No student should allow another person/s to use or copy from his/her academic output and present it as their own work;

Each student is required to attach a signed Declaration of Originality (see Annexures A and B) for each academic output submission (e.g. assignment, project, manuscript, dissertation and thesis); and Each student has the responsibility to request assistance from staff members should they require guidance and/or advice about plagiarism in their academic outputs.

Students have a moral obligation to report plagiarism incidents in academic and/or research environments. All whistle-blowers are protected in terms of the Policy on Prevention of Fraud, Corruption and Theft (Policy #: VCPOL010).

Plagiarism is a form of misconduct. The relevant part (Chapter 15 – Student Discipline) of the Prospectus, Part 1 (Students' rules and regulations) read as follows:

"Any student who contravenes the provisions of rule 15.1 of the disciplinary code is guilty of misconduct and will be dealt with in terms of the disciplinary code for students …

15.1.16 Handing in any written assignment for assessment in which the essential parts of the assignment have been copied from the work of another person, or any form of plagiarism."

All students must be fully aware that plagiarism offences/penalties can seriously affect their academic status and progress at TUT and other tertiary institutions. In the most serious cases, it can result in dismissal from the University and/or formal cancellation/retraction of current/previously submitted academic outputs. In addition, the University may indicate the nature and outcome of all plagiarism offences/penalties when it is required to provide a reference or conduct statement for the particular student.

35. Glossary

A "**Module coordinator**", is a lecturer responsible for coordinating the activities within the module. A module coordinator is responsible for setting up the study guide and ensuring that all the participants stay on schedule.

A "**Secondary module coordinator**", is a lecturer responsible for assisting the module coordinator with activities that scaled due to large student number quantities within the module.

A “**Site coordinator**”, is a lecturer responsible for coordinating the activities (within the module) on a learning site. The site coordinator is responsible for reporting all deviations from the activities scheduled to the module coordinator.

A “**Custodian department**”, is the department responsible for managing the module and all the resources required by the module.

ADDITIONAL DOCUMENTATION

Appendix A

(paste your additional notes here)

Appendix B

(paste your additional notes here)