College of VE – Future Technologies

COSC 2511: Introduction to Programming A

Programming Project

Due End of Week 15

Assessment weighting: This assessment is out of 100 and will contribute towards 30% of your overall course mark. (This is not a hurdle assessment)

Submission:

- This is a group project. Students must work in a group of 3. You must self-enrol into a group by the start of Week 11. After this time, groups will be locked.
 - You may choose who you would like to work with. If you do not have a group, then one will be assigned to you.
- Only one submission will be made per team.
- The due date for this assessment is the end of week 15.
- Submissions will be made via Canvas.
- AI TOOLS CANNOT BE USED IN THE COMPLETION OF THIS ASSESSMENT TASK.
- In this assessment task, you must not use any AI tools (excluding text editing software e.g., Grammarly) to generate any materials, content or ideas related to the task.
- After you have submitted your project, your team will be required to attend a mandatory meeting with your teacher to demonstrate your project.
 - o During this meeting you will be graded on your submission.
 - Your teacher will ask you questions to validate your understanding of the content written for the project. It is each member's responsibility to understand the project completely.
 - Marks can be deducted at the discretion of the teacher if you fail to answer questions asked.
 - IMPORTANT: If you do not attend this meeting, you will receive a mark of 0 for this assessment.

Assessment policy:

- Assessments submitted between 0-24 hours after the due date: 20% penalty.
- Assessments submitted between 24-48 hours after the due date: 50% penalty.
- Assessments submitted more than 48 hours after the due date will not be accepted.

What to Submit:

Your team should make one single submission in a .zip file named:

- GroupName+Number.zip
 - This will be based on the group name + number you have assigned yourselves on canvas.

Your zip file should include the following:

- Your team's design document containing all the requirements outlined in the section: What to complete
- Your team's completed java project including all required program files submitted as .java source code files. Keep the package/folder structure in your submission.

Extensions:

- Extensions will only be granted under exceptional circumstances and are intended to offer support and flexibility where unforeseen events have occurred preventing students from submitting projects on time.
- If an extension is required, project teams must apply via email to their lab teacher prior to the due date with an explanation of the unforeseen circumstances experienced.
- If an extension is granted by your teacher, it will be for a maximum of 7 calendar days.
- If further extensions are required, all project team members must individually apply for RMIT Special Consideration

Overview

This project is designed to give you an opportunity to exercise your algorithmic thinking and Java programming skills to solve programming problems with tools practiced throughout the semester in Introduction to Programming COSC2511.

You will demonstrate every topic learned throughout the semester in this assessment.

Team Composition:

You will form teams of 3 students in Canvas via the "People" link under the "Project Groups Tab". Select a group from your section and self -enrol along with the people you would like to work with.

If you are not sure how to access this, please ask your teacher in class.

You must allocate yourself to a team by the start of week 11. Any students not forming a team by this date will be allocated to a team.

What to Create (Core Requirements):

The topic of this project is that your submission should be a console-based text adventure game written in Java and should show an understanding of topics covered in Introduction to Programming COSC2511.

Full creative control is with your project team, and the story line is completely up to you! You might be exploring an alien planet, attacking a castle, or trying to make your way out of a spooky forest. Let your imagination run wild!

Your game environment should be based on a 5 x 5 grid layout (see visual example below) and should have a minimum of 9 locations that a player can visit throughout the game.

Your game must have a clear objective (to win the game) and potentially may have an alternate ending (optional) such as the player dying. Your game must include an inventory system allowing a player to pick up, carry, list and interact with a minimum of 5 items, and must include at least one interaction with a non-player character (npc).

Players must be able to navigate the game using a simple control system based on the four main points of a compass (north(n), south(s), east(e) & west(w)), and each time a location and any available items or interactions is visited a description of the location must be printed to the screen.

Once an item has been picked up, or a foe vanquished, subsequent visits to that location must omit the item or npc from the description (or in the case of a slain enemy, change its state to dead in the description).

Your goal is to display an understanding of the following programming tools used throughout your game.

What to complete

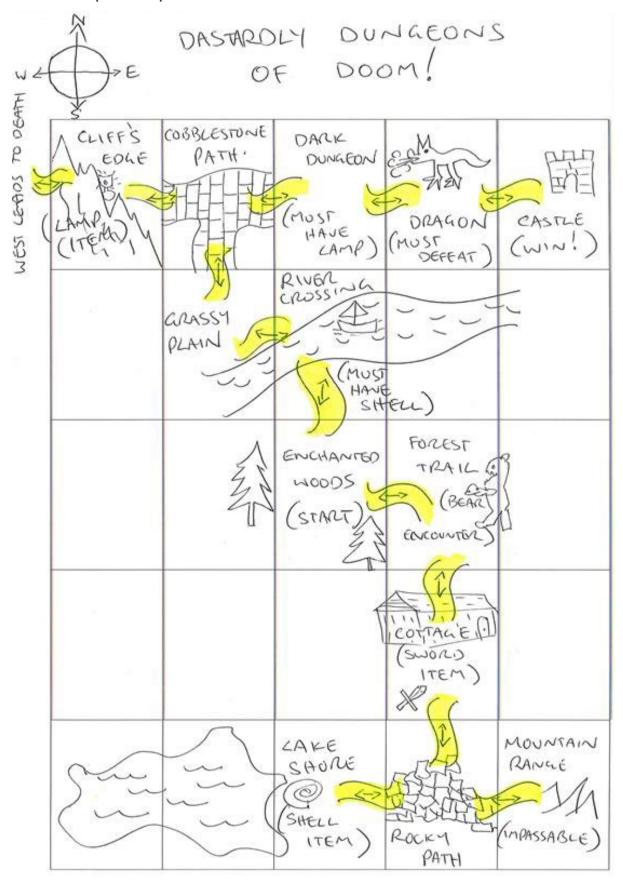
A comprehensive design document which includes:

- Overview of the game including:
 - o Objective
 - Gameplay Instructions
 - Map (see next page)
- Overview of each class created <u>OR</u> completed Javadoc outlining what each class does
- Algorithms for each method created <u>OR</u> completed Javadoc outlining what each method does
- A continuous bug report including how each issue was overcome
- Allocation of duties for each team member

A completed application which contains the following:

- Exemplary coding etiquette showing good indenting, consistent block bracing style, class naming conventions
- Appropriate use of comments.
- User (player) input
- Appropriate selection of and use of variables
- Selection statements including switch, if, else if and nested if & Implementation of user menu
- Iteration with the appropriate type of loops
- Use of random numbers
- Arrays
- Methods
- Classes

Game Map Example



Marking Guide – Documentation 35 % of the total mark

Game	3 Marks	2 Marks	1 Mark	0 Marks	
Objective	Clearly defined game	Demonstrated defined	Demonstrated game	Demonstrated no	
	objective that is	game objective present	objective present but	clearobjective	
	challenging but	but either too	not clear or achievable	defined to win the	
	achievable	challenging or too easy		game	
Comonloy	3 Marks	to achieve 1.5 Mark	0 Marks		
Gameplay Instructions	Clearly defined	Instructions provided	Instructions are not	-	
mondonono	instructions that are	provide too much/little	provided		
	concise but easy to	information	provided		
	follow				
Мар	5 Marks	3 Marks	2 Marks	1 Mark	0 Marks
	All the following	One of the following	Two of the following	Three of the	More than three
	criteria are met:	criteria is not met:	criteria are not met:	following criteria	of the following
	-Map created is a	-Map created is a 5X5	-Map created is a 5X5	are not met:	criteria are not
	5X5 grid.	grid.	grid.	-Map created is a	met:
	-All locations are clearly identified.	-All locations are clearly identified.	-All locations are clearly identified.	5X5 gridAll locations are	-Map created is a 5X5 grid.
	-Encounters and	-Encounters and items	-Encounters and items	clearly identified.	-All locations are
	items are clearly	are clearly identified.	are clearly identified.	-Encounters and	clearly identified.
	identified.	-All requirements are	-All requirements are	items are clearly	-Encounters and
	-All requirements are	clearly labelled (such	clearly labelled (such	identified.	items are clearly
	clearly labelled (such	as must have lamp)	as must have lamp)	-All requirements	identified.
	as must have lamp)			are clearly labelled	-All requirements
				(such as must	are clearly
				have lamp)	labelled (such as
Overview for	3 Marks	1.5 Marks	0 Marks		must have lamp)
classes	Each class file	Some of the class files	None of the class files	-	
o.uoooo	created has an	created have an	created have an		
	overview written	overview written	overview written		
	explaining its	explaining its purpose.	explaining its purpose.		
	purpose.				
		OR	OR		
	OR	0 (11 1 61	N. C.I. I. C.I.		
	Foob along file in the	Some of the class files	None of the class files		
	Each class file in the java doc has an	in the java doc has an overview written	in the java doc has an overview written		
	overview written	explaining its purpose.	explaining its purpose.		
	explaining its	oxpraning its purposer	oxpraning its purposer		
	purpose.				
Algorithms	14 Marks	11 Marks	8 Marks	4 Marks	0 Marks
	Algorithms or	Most algorithms or	Approximately half the	Less than half the	No appropriate
	Javadoc created for	Javadoc created for	algorithms created for	algorithms created	algorithms
	each Method inside	each Method inside	each Method inside	for each Method inside each class	created. OR
	each class (excluding	each class (excluding setters/getters/toString	each class (excluding setters/getters/toString	(excluding	Each algorithm
	setters/getters/toStri))	setters/getters/toS	created does not
	ng)	'	,	tring)	clearly addresses
	3,	Each algorithm created	Each algorithm created	<i>S</i> ,	the following:
	Each algorithm	clearly addresses the	clearly addresses the	Each algorithm	-requirements of
	created clearly	following:	following:	created clearly	each method
	addresses the	-requirements of each	-requirements of each	addresses the	clearly identified
	following:	method clearly identified	method clearly identified	following:	-all inputs,
	-requirements of each method clearly	-all inputs, outputs,	-all inputs, outputs,	-requirements of each method	outputs, and processing are
	identified	and processing are	and processing are	clearly identified	clearly identified
	-all inputs, outputs,	clearly identified	clearly identified	-all inputs,	, ,
	and processing are			outputs, and	OR
	clearly identified	OR	OR	processing are	Each method in
		Each method in the	Each method in the	clearly identified	the javadoc does
	OR	javadoc has a clear	javadoc has a clear	O.D.	not have a clear
	Each mathed in the	description which	description which	OR Each method in	description
	Each method in the javadoc has a clear	includes: -what the method does	includes: -what the method does	the javadoc has a	which includes: -what the method
	description which	-Requirements of each	-Requirements of each	clear description	does
	includes:	method clearly	method clearly	which includes:	-Requirements of
	-what the method	identified	identified	-what the method	each method
	does	- expected outputs,	- expected outputs,	does	clearly identified
	Doguiromonto of	and processing are	and processing are	-Requirements of	- expected
	-Requirements of	,		·	
	each method clearly identified	clearly identified	clearly identified	each method clearly identified	outputs, and

	- expected outputs,			- expected	processing are
	and processing are			outputs, and	clearly identified
	clearly identified			processing are	otourty labritinou
	otourty ruomanou			clearly identified	
Bug report	5 Marks	3 Marks	2 Marks	0 Marks	
	At least 10 bugs	At least 7 bugs	At least 4 bugs	Less than 4 bugs	
	identified during	identified during	identified during	identified during	
	development, each	development, each	development, each	development.	
	outlining the issue,	outlining the issue, who	outlining the issue, who		
	who discovered it,	discovered it, and a	discovered it, and a		
	and a solution	solution provided to	solution provided to		
	provided to rectify	rectify the issue	rectify the issue		
	the issue	,	,		
Allocation of	2 Marks	1 Mark	0 Marks		
Duties					
	All the following	One of the following	None of the following		
	criteria are met:	criteria is met:	criteria are met:		
	-Allocation of duties	-Allocation of duties for	-Allocation of duties for		
	for each member of	each member of the	each member of the		
	the group is clear.	group is clear.	group is clear.		
	-Workload is evenly	-Workload is evenly	-Workload is evenly		
	and distributed	and distributed	and distributed		

Marking Guide – Coding 65% of the total mark

Coding Style	5 Marks	3 Marks	0 Marks		
	All the following	Two of the	Less than two of		
	criteria are met:	following criteria	the following		
		are met:	criteria are met:		
	Code is indented				
	appropriately	Code is indented	Code is indented		
		appropriately	appropriately		
	Consistent block				
	bracing has been	Consistent block	Consistent block		
	used	bracing has been	bracing has been		
		used	used		
	Appropriate				
	naming	Appropriate	Appropriate		
	conventions used	naming	naming		
	for variables	conventions used	conventions used		
		for variables	for variables		
Commenting	5 Marks	3 Marks	2 Marks	0 Marks	
	At least 10	At least 7	At least 4	Less than 4 At	
	appropriate	appropriate	appropriate	appropriate	
	comments have	comments have	comments have	comments have	
	been used in the	been used in the	been used in the	been used in the	
	code	code	code	code	
User Input	6 Marks	4 Marks	3 Marks	2 Marks	0 Marks
	A range of user-	A range of user-	A range of user-	A range of user-	Little or no user
	inputs	inputs	inputs	inputs	input
	demonstrated	demonstrated	demonstrated	demonstrated	demonstrated
	showcasing at	showcasing at	showcasing at	showcasing at	OR
	least 3 datatypes	least 2 datatypes	least 2 datatypes	least 1 datatype	Approx. less
	accepted.	accepted.	accepted.	accepted.	than half the
	AND	AND	AND	AND	user inputs are
	All user inputs are	Most user inputs	Approx. half the	Approx. half the	appropriate and
	appropriate and	are appropriate	user inputs are	user inputs are	have appropriate
	have appropriate	and have	appropriate and	appropriate and	data validation
	data validation	appropriate data	have appropriate	have appropriate	techniques
	techniques applied	validation	data validation	data validation	applied
		techniques applied	techniques applied	techniques	
				applied	
Datatypes	6 Marks	4 Marks	2 Marks	0 Marks	
	Demonstrated at	Demonstrated at	Demonstrated at	Demonstrated	
	least 4 appropriate	least 3 appropriate	least 2 appropriate	less than 2	
	data types used	data types used	data types used	appropriate data	
	showing an	showing an	showing an	types used.	
	understanding of	understanding of	understanding of		
	these data types	these data types	these data types		
	written in the code	written in the code	written in the code		

Selection	8 Marks	6 Marks	3 Marks	0 Marks	
Statements	At least 10	At least 7 selection	At least 4 selection	Less than 4	
	selection	statements used.	statements used.	selection	
	statements used.	AND	AND	statements used	
	AND	Appropriate use of	Appropriate use of	in the code	
	Appropriate use of	selection	selection	OR	
	selection statements that	statements that cover each of the	statements that cover at least 3 the	Use of selection statements that	
	cover each of the	following:	following:	cover less than 3	
	following:			of the following:	
		-If-else	-If-else		
	-If-else	-Switch-Case	-Switch-Case	-If-else	
	-Switch-Case	-Nesting	-Nesting	-Switch-Case	
	-Nesting -Implementation of	-Implementation of user menu	-Implementation of user menu	-Nesting -Implementation	
	user menu	usei illellu	usei illellu	of user menu	
Iteration	8 Marks	6 Marks	3 Marks	0 Marks	
	At least 8 different loops used	At least 6 different loops used	At least 4 different loops used	Less than 4 different loops	
	appropriately	appropriately	appropriately	used	
	AND	AND	AND	appropriately	
	Appropriate	Appropriate	Appropriate	OR	
	selection of loops	selection of loops	selection of loops	Only one of the	
	applied including	applied including	applied including	following loops	
	the use of:	the use of:	the use of any 2:	applied:	
	-while -for	-while -for	-while -for	-while -for	
	-do-while	-do-while	-do-while	-do-while	
Random	3 Marks	2 Marks	1 Mark	0 Marks	
Numbers	At least 3 Random	At least 2 Random	At least 1 Random	No Random	
	numbers are	numbers are	number is	number is	
	generated and	generated and	generated and	generated	
	used appropriately	used appropriately	used appropriately		
	in the code	in the code	in the code	0.14	
Arrays	6 Marks At least 4 different	4 Marks At least 3 different	2 Marks At least 2 different	0 Marks Less than 2 arrays	
	arrays created with	arrays created with	arrays created with	created with	
	appropriate	appropriate	appropriate	appropriate	
	datatypes	datatypes	datatypes	datatypes	
	AND	AND	AND	Or	
	Appropriate array	Appropriate array	Appropriate array	Less than 2 arrays	
	type selected	type selected	type selected	use appropriate	
	(standard/dynamic) AND	(standard/dynamic) AND	(standard/dynamic) AND	array type.	
	Both:	Both:	One of:		
	-Standard Arrays	-Standard Arrays	-Standard Arrays		
	-Standard Arrays				
	-Array Lists	-Array Lists	-Array Lists		
	-Array Lists are used in the	-Array Lists are used in the	are used in the		
Methods	-Array Lists are used in the code	-Array Lists are used in the code	are used in the code	2 Marks	0 Marks
Methods	-Array Lists are used in the	-Array Lists are used in the	are used in the	2 Marks At least 4	0 Marks Less than 4
Methods	-Array Lists are used in the code 10 Marks	-Array Lists are used in the code 8 Marks	are used in the code 5 Marks		
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have	are used in the code 5 Marks At least 4 different appropriate methods have	At least 4 different appropriate	Less than 4 different appropriate
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created	are used in the code 5 Marks At least 4 different appropriate methods have been created	At least 4 different appropriate methods have	Less than 4 different appropriate methods have
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main,	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding	At least 4 different appropriate methods have been created	Less than 4 different appropriate methods have been created
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and	At least 4 different appropriate methods have been created (excluding	Less than 4 different appropriate methods have been created (excluding
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main,	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding	At least 4 different appropriate methods have been created	Less than 4 different appropriate methods have been created
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods)	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods)	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods)	At least 4 different appropriate methods have been created (excluding accessor and	Less than 4 different appropriate methods have been created (excluding accessor and
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND At least 4 different	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 3 different	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 2 different	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR At least 2	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR Less than 2
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND At least 4 different	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 3 different	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 2 different	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR At least 2 different return	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR Less than 2 different return
	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND At least 4 different return types used	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 3 different return types used	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 2 different return types used	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR At least 2 different return types used	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR Less than 2
Methods	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND At least 4 different return types used	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 3 different return types used	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 2 different return types used	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR At least 2 different return types used 0 Marks	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR Less than 2 different return
	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND At least 4 different return types used 8 Marks (Excluding the	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 3 different return types used 6 Marks (Excluding the	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 2 different return types used 3 Marks (Excluding the	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR At least 2 different return types used 0 Marks (Excluding the	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR Less than 2 different return
	-Array Lists are used in the code 10 Marks At least 10 different appropriate methods have been created (excluding main, accessor and mutator methods) AND At least 4 different return types used	-Array Lists are used in the code 8 Marks At least 7 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 3 different return types used	are used in the code 5 Marks At least 4 different appropriate methods have been created (excluding accessor and mutator methods) AND At least 2 different return types used	At least 4 different appropriate methods have been created (excluding accessor and mutator methods) OR At least 2 different return types used 0 Marks	Less than 4 different appropriate methods have been created (excluding accessor and mutator methods) OR Less than 2 different return

least 3 additional	least 2 additional	least 1 additional	additional
classes created.	classes created.	class created.	classes created.
Each of these	Each of these	Each of these	
classes has at least	classes has at least	classes has at least	
one object	one object	one object	
instantiated.	instantiated.	instantiated.	