Assessment Task Portfolio of Evidence

ICTPRG547_AT2_PE_TQM_V1.docx

Student Name		**Student Number**	
Unit Code/s & Name/s	ICTPRG547 Apply advanced programming skills in another language		
Cluster Name *If applicable*	N/A		
Assessment Name	Java - Programming Portfolio of Evidence	**Assessment Task No.**	2 of 2
Assessment Due Date	Week 16	**Date Submitted**	/ /
Assessor Name			
Student Declaration: I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature.			
Student Signature		**Date**	/ /

Instructions to Student	**General Instructions:** You are employed by ***Uptown IT's Software Solutions Department*** as a Programmer. You have been assigned to a new project and your task is develop a software solution for your client. Your teacher/assessor will take on the role of the Project Manager assigned to this project by Uptown IT. Read the project documentation provided and familiarise yourself with the Project Scenario or Case Study before proceeding with project tasks. Confirm anything you are not sure about the project with the Project Manager (teacher/assessor). It is essential that you have a clear understanding of the scenario and tasks that you need to complete. This assessment instrument requires the student to complete a programming project that is divided into five (5) parts: – PART 1. Environment setup – PART 2. Build the application – code the solution – PART 3. Program documentation

- PART 4, Debugging, testing and optimising code
- PART 5. Contingency and advanced programming concepts

Materials Required:

- Students are required to provide their own storage device. The recommendation for this qualification is an external SSD drive with at least 256 GB capacity, if you need to store a copy of the Virtual Machine (VM). For assessment files only, a 64 GB thumb drive will be sufficient.
- Access to PCs and peripherals – these may differ between classrooms
- Access to Internet
- Access to Connect (LMS)
- Access to Integrated Development Environment (IDE)
- Access to Word processing software, such as Microsoft Word
- Access to special purpose tools, equipment and materials to complete the assessment, for example diagramming software.

Online Delivery:

- Student to supply their own PC or laptop an internet access

Documentation:

- Uptown IT Scenario Requirements
- CD_AchivePrototype_SampleData.txt
- CD_AchivePrototype.mdb
- UptownIT_Java_Coding_Standards.docx
- UptownIT_External_Documentation_Template.docx
- UptownIT_Test_Cases_Report_Template.xlsx

Assessment Criteria:
To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate the following key skills/tasks/knowledge to an acceptable industry standard. Demonstrated ability to:

- Determine the information and documentation required to develop and document the project
- Establish and articulate project specifications with Project Manager
- Code software solution from specifications
- Code using advanced data structures and algorithms
- Design, document and implement dynamic data structures including:
    - Double-linked lists
    - Binary trees
- Design algorithm and constructs
- Code use hashing techniques
- Code advanced searching and sorting techniques

- Use third-party libraries and inter-process language communication and signals
- Code application to work with GUI
- Use debugging tools in-build in the IDE as well as third party stand-alone tools
- Detect and resolve logical, syntactical and design problems
- Create application as per specifications
- Debug application by applying a tracer, breakpoints, and associated watch tools.
- Design, carry out and document tests to ensure that application conforms to requirements
- Prepare internal and external documentation following organisational standards
- Complete project documentation and seek approval from Project manager

Refer to the marking criteria for specific details:
ICTPRG547_AT2_MC_TQM_V1

Details of location:
Skills and knowledge in this competency unit must be demonstrated in a workplace or simulated environment where the conditions are typical of those in a working environment in this industry.

Research and programming activities may be conducted in the classroom or at home.

If you are unable to attend a scheduled assessment activity, you must notify your teacher before the assessment is due and supply a doctor's certificate and approval from the team manager for an extension.

Time restrictions:
This portfolio is designed to take place over 5 weeks and approximately 20 hours. The student is expected to attend classes as per timetable details and should be able to commit up to 3 hours per week of their own time to study or study related activities.

Interactions:
Teamwork skills are essential in the IT industry therefore you should work in teams to consult and collaborate on practical activities. However, each student must complete the assessment tasks individually (unless indicated).

Level of assistance permitted:
Staff cannot directly show students answers or solutions but support and guide them to complete tasks individually. Teachers and tutors should be available in class, and accessible by email for students working from home.

Reasonable Adjustments:
A reasonable adjustment is available to students for a variety of reasons, including: disability, language, literacy and numeracy (LLN) problems or

	extenuating circumstances. Talk to your teacher, counsellor or disability officer if you require extra support or an extension based on the conditions identified. **Number of Attempts:** You will receive up to two (2) attempts at this assessment task. Should your 1st attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant sections / questions with you and will arrange a due date for the submission of your 2nd attempt. If your 2nd submission is unsatisfactory (U), or you fail to submit a 2nd attempt, you will receive an overall unsatisfactory result for this assessment task. Only one re-assessment attempt may be granted for each assessment task. ***For more information, refer to the Student Rules.*** **Work, Health and Safety:** The work environment should be assessed for safety prior to class. Special consideration should be taken regarding potential ICT related hazards such as tripping hazards, electromagnetic radiation, ergonomics and posture. TAFE Queensland health and safety policies and procedures should be followed at all times.
Submission details	**Evidence Required to be Submitted:** Insert your details on the cover page and sign the Student Declaration. Include this template with your submission. **Submission via Connect:** Upload a single file into Assessment 2 (AT2) Assignment Folder in Connect. Multiple files can be compressed into a single file. ***ICTPRG547_AT2_PE_ Surname_StudentNumber*** TAFE Queensland Learning Management System: **Accessing Connect:** Connect URL: https://connect.tafeqld.edu.au/d2l/login Username: 9 digit student number Password: <your password> For password reset go to: https://passwordreset.tafeqld.edu.au/default.aspx
Instructions for the Assessor	**Online Delivery:** Please revise and modify the Instructions to Student section if you are delivering online. **Assessment Conditions:** Skills and knowledge in this competency unit must be demonstrated in a workplace or simulated environment where the conditions are typical of those in a working environment in this industry.

To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide.

Gather evidence to demonstrate consistent performance in conditions that are safe and replicate the workplace. Noise levels, production flow, interruptions and time variances must be typical of those experienced in the programming and software development field of work and include access to:

- software development environment
- technical requirements

Specifications of assessment:
Ensure that students read and familiarise themselves with the Project Scenario and the provided relevant files and/or resources before attempting the assessment.

Storage Devices:

- Students are required to provide their own storage device. The recommendation for this qualification is an external SSD drive with at least 256 GB capacity, if you need to store a copy of the Virtual Machine (VM). For assessment files only, a 64 GB thumb drive will be sufficient.

Assessor to Provide:

- Access to lab PCs and peripherals – these may differ between classrooms
- Access to Internet
- Access to Connect (LMS)
- Access to Integrated Development Environment (IDE)
- Access to Word processing software, such as Microsoft Word.
- Access to special purpose tools, equipment and materials to complete the assessment, for example diagramming software.

Online Delivery:

- Student to supply their own PC or laptop an Internet Access

Documentation:

- Uptown IT Scenario Requirements
- CD_AchivePrototype_SampleData.txt
- CD_AchivePrototype.mdb
- UptownIT_Java_Coding_Standards.docx
- UptownIT_External_Documentation_Template.docx
- UptownIT_Test_Cases_Report_Template.xlsx

Level of Assistance Permitted:

Teachers and tutors should be available in class, and accessible by email for students working from home. Staff cannot directly show students answers or solutions to assessment but support and guide them to complete tasks individually. Students with disability will receive reasonable adjustments.

	Interactions: Teamwork skills are essential in the IT industry therefore you should work in teams to consult and collaborate on practical activities. However, each student must complete the assessment tasks individually (unless indicated). **Contingencies:** A reasonable adjustment is available to students for a variety of reasons, including: disability, language, literacy and numeracy (LLN) problems or extenuating circumstances. **Work, Health and Safety:** The work environment should be assessed for safety prior to class. Special consideration should be taken regarding potential ICT related hazards such as tripping hazards, electromagnetic radiation, ergonomics and posture. TAFE Queensland health and safety policies and procedures should be followed at all times.
Note to Student	An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide.

Note: Attendance and Authenticity

The teacher needs to be able to review the progress of your work throughout the semester, and therefore requires an appropriate level of attendance in class. Where the authenticity of your submitted work may be uncertain, the teacher reserves the right to call for an in-person review or an additional written test.

Scenario Script and Requirements

You are employed by ***Uptown IT's Software Solutions Department*** as a Programmer. You have been assigned to a new project and your task is develop a CD Archive Prototype software solution for your client.

Your teacher/assessor will take on the role of the Project Manager assigned to this project by Uptown IT.

The project requires you to build a prototype system to emulate the management of a CD Archive collection, utilising an automated robotic system, linked via a network. A screen design of this prototype system has been prepared and presented below and on the following pages.

The anticipated benefit of the system is to better manage the CD archive and to save time with:

- **Adding** a new CD to the collection,
- **Retrieving** a CD from and **returning** a CD back into the system,
- **Removing** a CD from the system,
- Periodically **sorting** the CD collection.

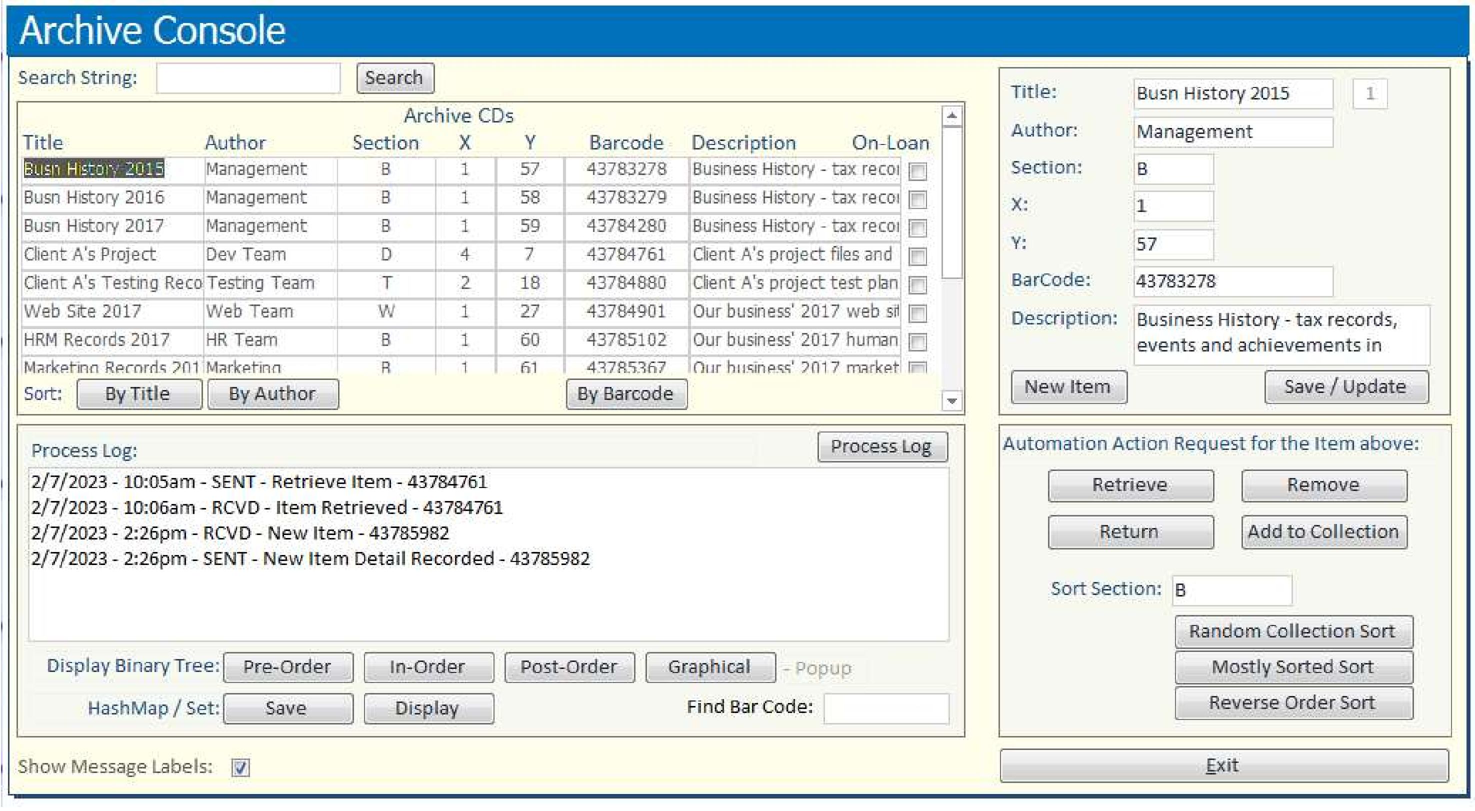
Note: the collection will be separated into specific **sections**.

Title: Assorted DVD case lot, Photo by Sidney Pearce
https://unsplash.com/search/photos/cds

Your solution will provide the required console management module and a prototype of the 'automation' component of the system.

A sample **design prototype** of your solution has been prepared. Sample screens and explanatory notes are provided on the following pages.

The top left of the screen below – the **Archive Console** - displays a list of CDs currently in the collection. The top right displays details for the currently selected CD. The bottom left displays a log of messages sent to and from the Automation Console (shown on the next page). The bottom right provides buttons for each of the potential automation options.



Archive CDs

Title	Author	Section	X	Y	Barcode	Description	On-Loan
Busn History 2015	Management	B	1	57	43783278	Business History - tax recoi	☐
Busn History 2016	Management	B	1	58	43783279	Business History - tax recoi	☐
Busn History 2017	Management	B	1	59	43784280	Business History - tax recoi	☐
Client A's Project	Dev Team	D	4	7	43784761	Client A's project files and	☐
Client A's Testing Reco	Testing Team	T	2	18	43784880	Client A's project test plan	☐
Web Site 2017	Web Team	W	1	27	43784901	Our business' 2017 web si	☐
HRM Records 2017	HR Team	B	1	60	43785102	Our business' 2017 human	☐
Marketing Records 201	Marketing	B	1	61	43785367	Our business' 2017 market	☐

Title:	Busn History 2015	1
Author:	Management	
Section:	B	
X:	1	
Y:	57	
BarCode:	43783278	
Description:	Business History - tax records, events and achievements in	

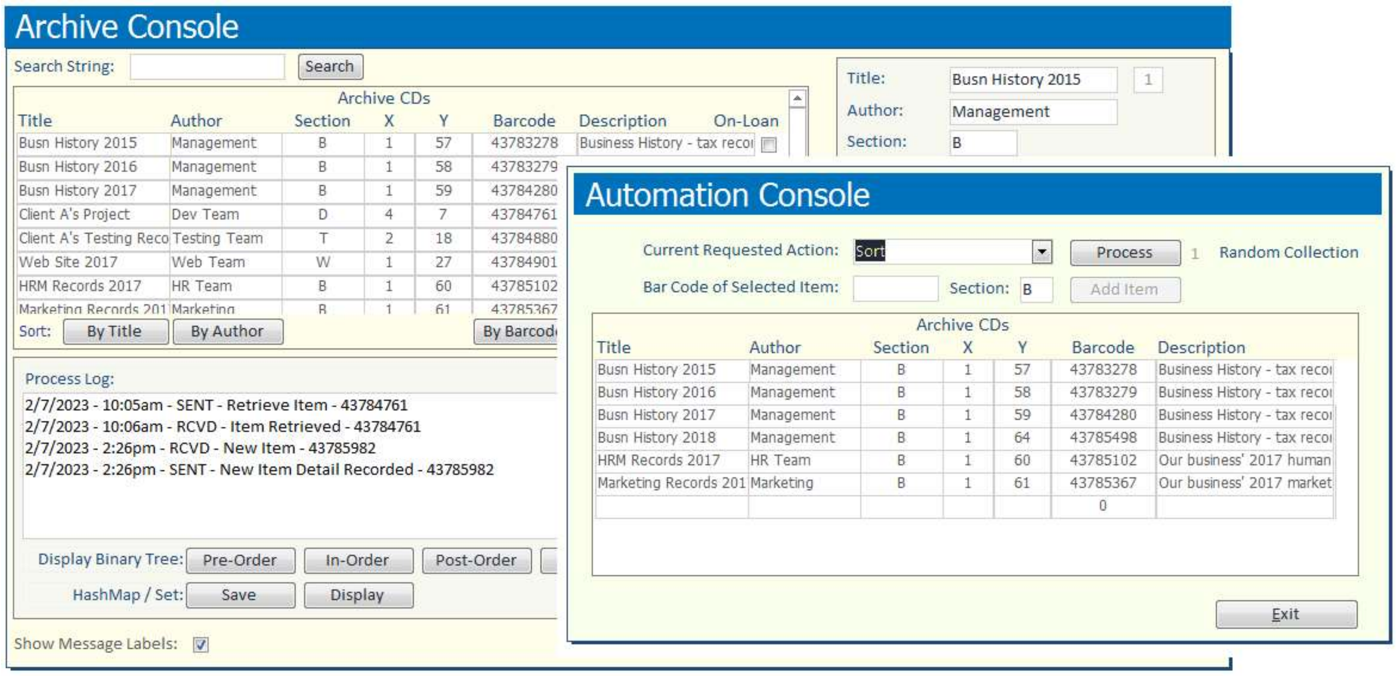
Process Log:

```
2/7/2023 - 10:05am - SENT - Retrieve Item - 43784761
2/7/2023 - 10:06am - RCVD - Item Retrieved - 43784761
2/7/2023 - 2:26pm - RCVD - New Item - 43785982
2/7/2023 - 2:26pm - SENT - New Item Detail Recorded - 43785982
```

Sort Section: B

Show Message Labels: ☑

After selecting an action on the Archive Console, the **Automation Console** will be displayed and echo the request. This Automation Console would eventually be replaced with a controller for managing the automation of a robotic arm.



Archive Console

Search String: [ ] Search

Archive CDs

Title	Author	Section	X	Y	Barcode	Description	On-Loan
Busn History 2015	Management	B	1	57	43783278	Business History - tax recoi	☐
Busn History 2016	Management	B	1	58	43783279		
Busn History 2017	Management	B	1	59	43784280		
Client A's Project	Dev Team	D	4	7	43784761		
Client A's Testing Reco	Testing Team	T	2	18	43784880		
Web Site 2017	Web Team	W	1	27	43784901		
HRM Records 2017	HR Team	B	1	60	43785102		
Marketing Records 201	Marketing	B	1	61	43785367		

Title: Busn History 2015 1
Author: Management
Section: B

Process Log:

```
2/7/2023 - 10:05am - SENT - Retrieve Item - 43784761
2/7/2023 - 10:06am - RCVD - Item Retrieved - 43784761
2/7/2023 - 2:26pm - RCVD - New Item - 43785982
2/7/2023 - 2:26pm - SENT - New Item Detail Recorded - 43785982
```

Show Message Labels: ☑

Sort: By Title By Author By Barcode
Display Binary Tree: Pre-Order In-Order Post-Order
HashMap / Set: Save Display

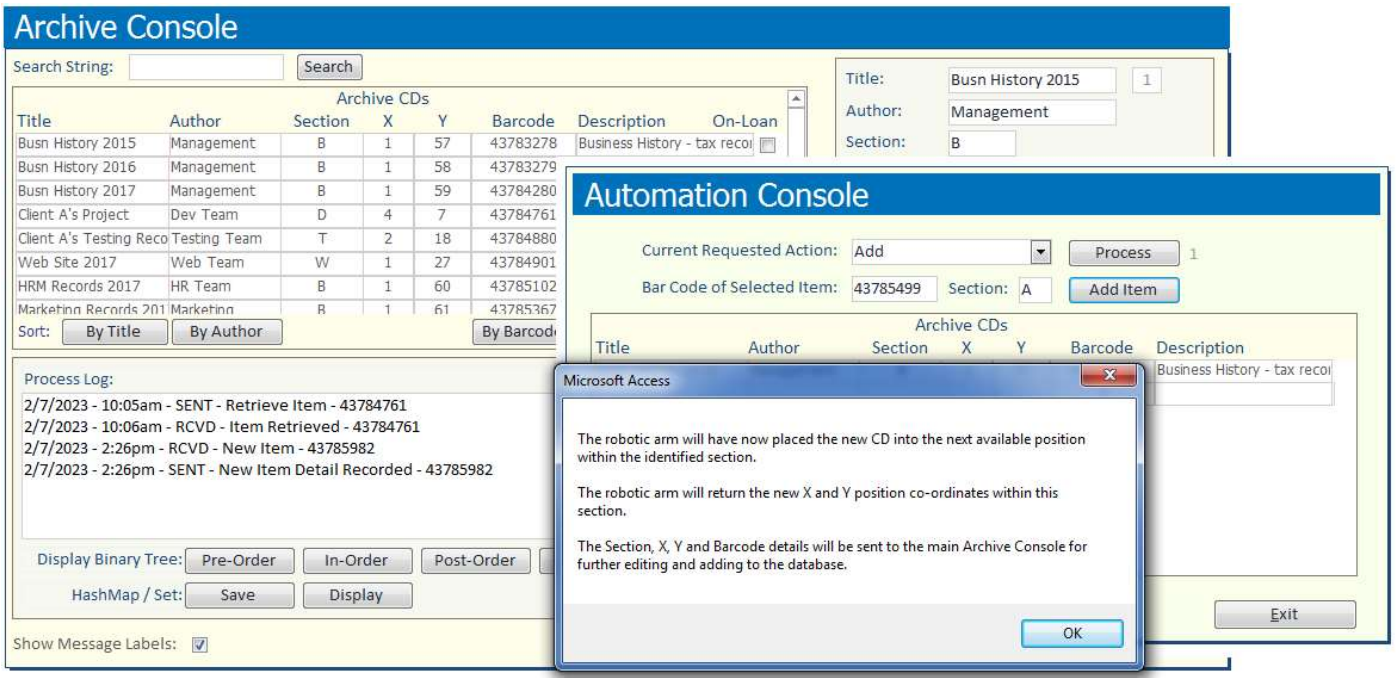
Automation Console

Current Requested Action: Sort
Bar Code of Selected Item: [ ] Section: B
1 Random Collection

Archive CDs

Title	Author	Section	X	Y	Barcode	Description
Busn History 2015	Management	B	1	57	43783278	Business History - tax recoi
Busn History 2016	Management	B	1	58	43783279	Business History - tax recoi
Busn History 2017	Management	B	1	59	43784280	Business History - tax recoi
Busn History 2018	Management	B	1	64	43785498	Business History - tax recoi
HRM Records 2017	HR Team	B	1	60	43785102	Our business' 2017 human
Marketing Records 201	Marketing	B	1	61	43785367	Our business' 2017 market
					0	

On clicking the **Process** button (and **Add Item** button if required) within the design prototype, a popup message will be displayed outlining the anticipated automation required.



Archive Console

Search String: [ ] Search

Archive CDs

Title	Author	Section	X	Y	Barcode	Description	On-Loan
Busn History 2015	Management	B	1	57	43783278	Business History - tax recoi	☐
Busn History 2016	Management	B	1	58	43783279		
Busn History 2017	Management	B	1	59	43784280		
Client A's Project	Dev Team	D	4	7	43784761		
Client A's Testing Reco	Testing Team	T	2	18	43784880		
Web Site 2017	Web Team	W	1	27	43784901		
HRM Records 2017	HR Team	B	1	60	43785102		
Marketing Records 201	Marketing	B	1	61	43785367		

Title: Busn History 2015 1
Author: Management
Section: B

Process Log:

```
2/7/2023 - 10:05am - SENT - Retrieve Item - 43784761
2/7/2023 - 10:06am - RCVD - Item Retrieved - 43784761
2/7/2023 - 2:26pm - RCVD - New Item - 43785982
2/7/2023 - 2:26pm - SENT - New Item Detail Recorded - 43785982
```

Show Message Labels: ☑

Sort: By Title By Author By Barcode
Display Binary Tree: Pre-Order In-Order Post-Order
HashMap / Set: Save Display

Automation Console

Current Requested Action: Add
Bar Code of Selected Item: 43785499 Section: A
1

Archive CDs

Title	Author	Section	X	Y	Barcode	Description
						Business History - tax recoi

Microsoft Access

The robotic arm will have now placed the new CD into the next available position within the identified section.

The robotic arm will return the new X and Y position co-ordinates within this section.

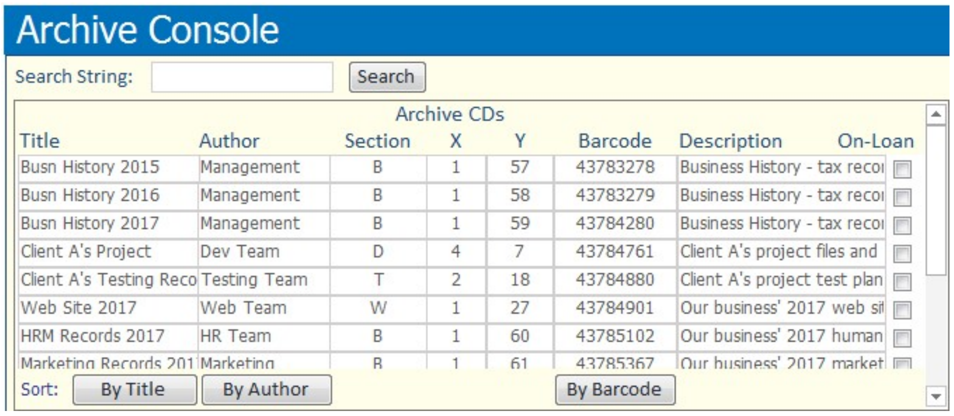
The Section, X, Y and Barcode details will be sent to the main Archive Console for further editing and adding to the database.

Assessment Instructions are as follows:

In relation to the CD list - top left of the **Archive Console** – you able to search for a CD using the **Search** facility at the top and **sort** the list using the 3 sort buttons at the bottom of the list.

Each CD has:

- a **Title**
- an **Author**
- a **Section** that it is to be stored in
- **X** and **Y** coordinates within that section
- a **Barcode** located on the end of each CD case
- a **Description** of the CD content
- an **On-loan** check box – EG: *Yes* – currently on loan and out of the collection.



Archive Console

Search String: [ ] Search

Archive CDs

Title	Author	Section	X	Y	Barcode	Description	On-Loan
Busn History 2015	Management	B	1	57	43783278	Business History - tax recoı	☐
Busn History 2016	Management	B	1	58	43783279	Business History - tax recoı	☐
Busn History 2017	Management	B	1	59	43784280	Business History - tax recoı	☐
Client A's Project	Dev Team	D	4	7	43784761	Client A's project files and	☐
Client A's Testing Reco	Testing Team	T	2	18	43784880	Client A's project test plan	☐
Web Site 2017	Web Team	W	1	27	43784901	Our business' 2017 web sit	☐
HRM Records 2017	HR Team	B	1	60	43785102	Our business' 2017 human	☐
Marketing Records 201	Marketing	B	1	61	43785367	Our business' 2017 market	☐

Sort: By Title | By Author | By Barcode

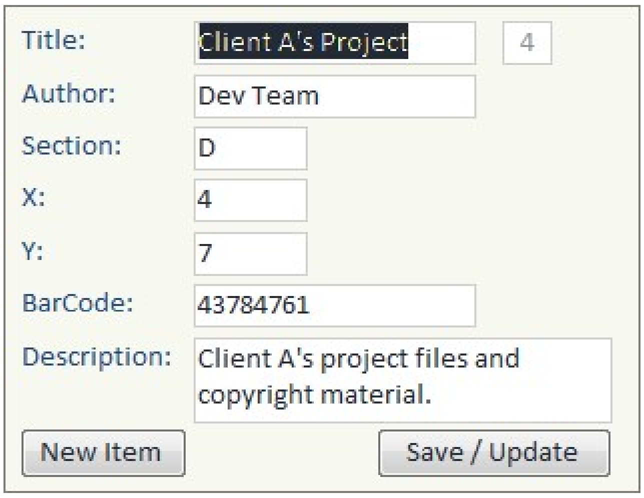
With reference to the CD list, you are required to program the following:

- A **user interface** similar to the design provided. A JTable or equivalent is required.
- The **reading of data from a data file** such as **CD_ArchivePrototype_SampleData.txt**, and loading this data into an **ArrayList** before displaying on screen within your table.
- A **search** facility to filter your displayed data.
- **Three sort buttons**. Each button must utilise a fully coded sort routine (not from a java or third-party library) and **each sort routine must be based on a different sorting algorithm**.

When the operator **clicks on one of the CDs** on the left of the screen, the CD's full detail is displayed within the JTextFields on the right of the screen.

The **New Item** button will clear these text fields and allow the operator to enter the details of a new item being added to the collection.

Once the details of the new item have been entered, the **Save/Update** button will save the entry to the ArrayList, and the full ArrayList to the data file (or its equivalent, eg: database).

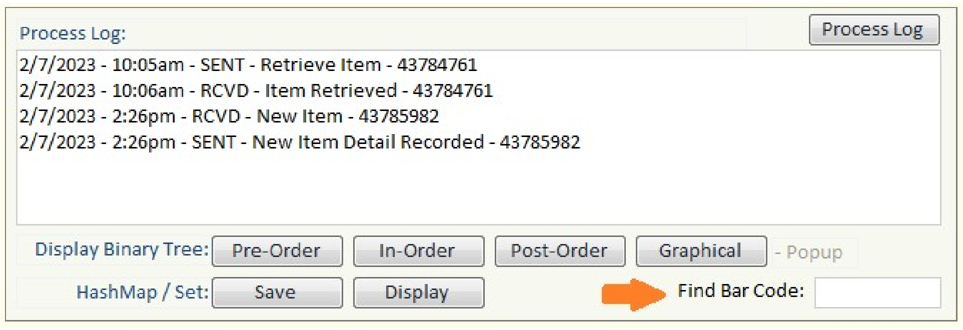


Title:	Client A's Project	4
Author:	Dev Team	
Section:	D	
X:	4	
Y:	7	
BarCode:	43784761	
Description:	Client A's project files and copyright material.	

On the bottom right of the screen, by default, the automation requests and associated messages to and from the **Automation Console** are displayed in a **Process Log** text area.

The first row of buttons at the bottom of the text area will display a **Binary Tree** of the CDs - **Barcode and then Title** of each CD:

- **Pre-Order** – displays the binary tree in accordance with a Pre-Order traversal.
- **In-Order** – displays the binary tree in accordance with an In-Order traversal.
- **Post-Order** – displays the binary tree in accordance with a Post-Order traversal.
- **Graphical** – displays a 2D graphical representation of the Binary tree – on a popup screen – sorted by and displaying only Barcodes.



Process Log:

```
2/7/2023 - 10:05am - SENT - Retrieve Item - 43784761
2/7/2023 - 10:06am - RCVD - Item Retrieved - 43784761
2/7/2023 - 2:26pm - RCVD - New Item - 43785982
2/7/2023 - 2:26pm - SENT - New Item Detail Recorded - 43785982
```

You are required to **fill the Binary Tree after clicking the '<u>By Title</u>' sort button** on the top left of the screen.

The **Save** button within the second row of buttons on the bottom left of the screen will save a **HashMap** or **HashSet** of the Binary Tree data.

The **Display** button will display the HaspMap or HashSet in the text area.

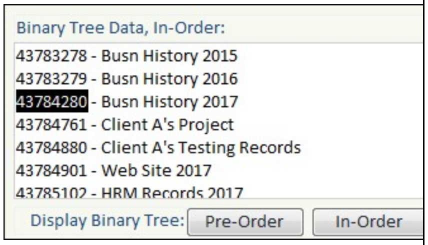
The **Process Log** button on the top right of the text area will re-display the Process Log data.

NOTE: The Process Log data is to be held in memory and managed using a **Doubly Linked List**.

As noted below, you are also required to add a feature that responds to an ***Operating System Signal****. And to add a* ***Third Party Library*** *and utilise it at least once within your project.* Two or more ***optimisation techniques*** need to be applied to your code.

The **Find Bar Code:** text field (on the right) can be used by the operator to locate a specific entry within the 'Linked List' when the Linked List is displayed, and within the Binary Tree when a Binary Tree listing is displayed. The associated full entry detail will then be displayed on the top right of the main screen.

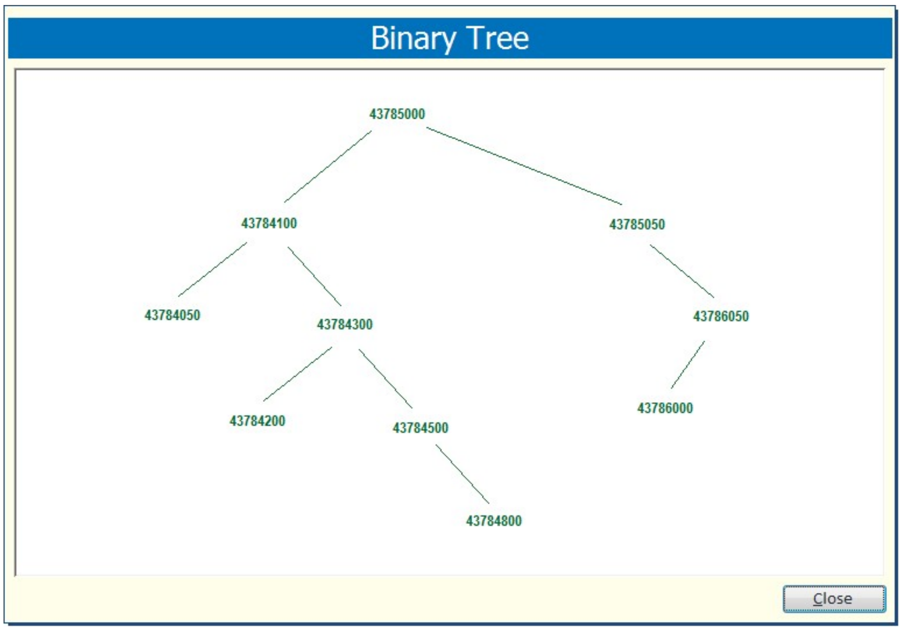
In-Order Binary Tree listing:



Binary Tree Data, In-Order:

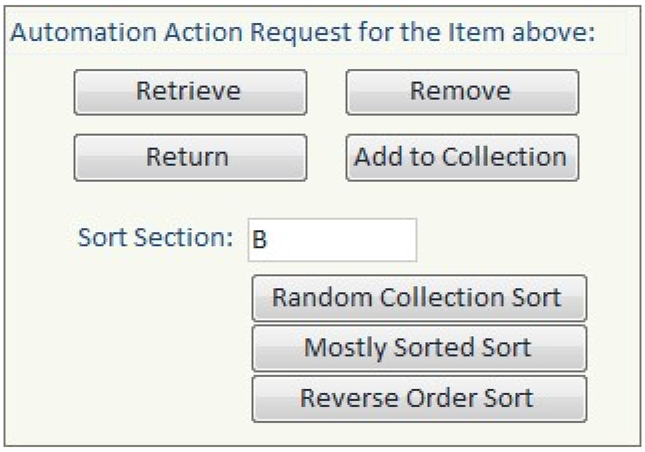
```
43783278 - Busn History 2015
43783279 - Busn History 2016
43784280 - Busn History 2017
43784761 - Client A's Project
43784880 - Client A's Testing Records
43784901 - Web Site 2017
43785102 - HRM Records 2017
```

NOTE: In relation to the **2D graphical Binary Tree** (as shown right), you are **not** required to prepare this in order to achieve a satisfactory grade for this assessment.



The various buttons on the bottom right of the screen are used to manage the automation requests.

These requests are detailed on the following pages.

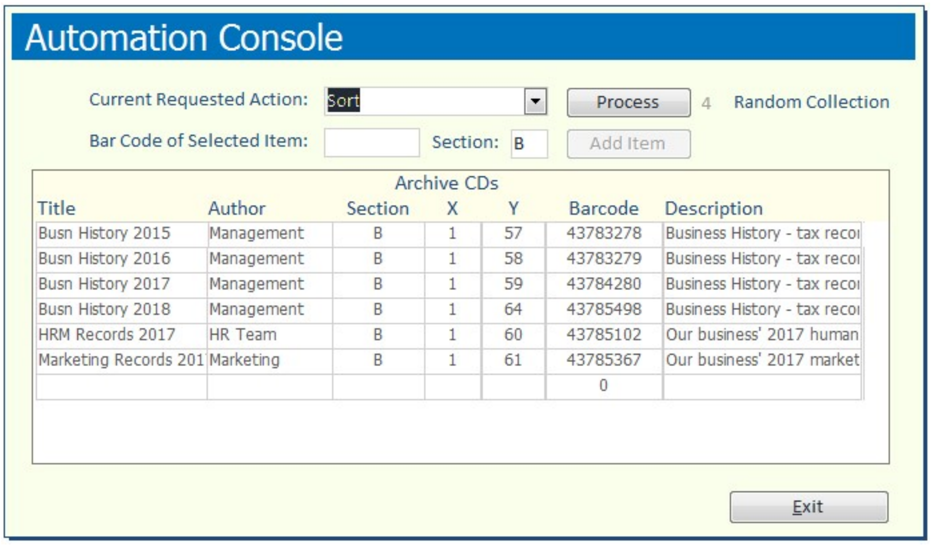


The automation requests are to be sent across a network to an Automation Console (shown right).

This Console will emulate the actions of a robotic arm. Click the **Process** and **Add Item** buttons to review further details.

Messages will need to be sent both to and from the Automation Console and will need to be coded in a manner that will allow each Console to follow-up on (action) requests as required.

Note: Add, Remove, Retrieve and Return actions are specific to a single CD. The Sort actions are applicable to a given section of CDs.

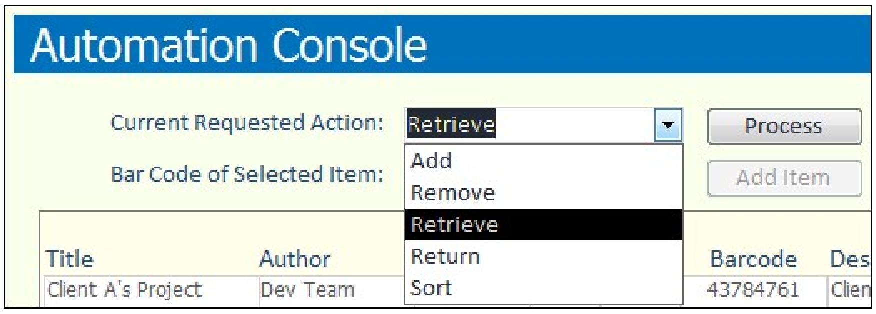

Automation Console

Current Requested Action: Sort | Process | 4 Random Collection

Bar Code of Selected Item: | Section: B | Add Item

Archive CDs

Title	Author	Section	X	Y	Barcode	Description
Busn History 2015	Management	B	1	57	43783278	Business History - tax recoı
Busn History 2016	Management	B	1	58	43783279	Business History - tax recoı
Busn History 2017	Management	B	1	59	43784280	Business History - tax recoı
Busn History 2018	Management	B	1	64	43785498	Business History - tax recoı
HRM Records 2017	HR Team	B	1	60	43785102	Our business' 2017 human
Marketing Records 201	Marketing	B	1	61	43785367	Our business' 2017 market
					0	


Automation Console

Current Requested Action: Retrieve

Bar Code of Selected Item:

Title	Author	Barcode	Des
Client A's Project	Dev Team	43784761	Clien

ADDITIONALLY: Your program needs to utilise a **third party library** plus the management of an **operating system signal** as part of its implementation. Two or more **optimisation techniques** need to be applied to your code.

With respect to the Automation Console Process options, the following actions would be automated:

If the action = ‘**Add’** then

- A new CD will have been handed to the robotic arm.
- When the 'Process' button is pressed, the robotic arm will read and return the new Barcode.
- The Barcode will be entered into the 'Barcode' field.
- Focus will be given to the 'Section' text field so the operator can indicate which section the CD needs to be added to.
- The 'Add Item' button will be enabled.

THEN on clicking the **Add Item** button:

- The robotic arm will have now placed the new CD into the next available position within the identified section.
- The robotic arm will return the new X and Y position co-ordinates within this section.
- The Section, X, Y and Barcode details will be sent to the main Archive Console for further editing and adding to the database.

If the action = ‘**Remove’** then

- The 'Remove' request will send commands to the robotic arm to find the relevant CD and make it available to the operator.
- Once the CD is found, a message is sent back to the main Archive Console.
- The main Archive Console will then remove the CD from the collection.

THEN a follow-up message:

- The CD is available for collection, and has been removed from the database.

If the action = ‘**Retrieve’** then

- The 'Retrieve' request will send commands to the robotic arm to find the relevant CD and make it available to the operator.
- Once the CD is found, a message is sent back to the main Archive Console.
- The main Archive Console will then note that the CD is currently out of the collection.
- The CD is available for collection, and has been noted as 'On-Loan' from the collection.

If the action = ‘**Return’** then

- A CD will have been handed to the robotic arm.
- When the 'Process' button is pressed, the robotic arm will place the CD back into its recorded position within the collection.
- A message will then be sent back to the main Archive Console.
- The main Archive Console will then note that the CD has been returned to the collection.

If the action = ‘**Sort’** then

- Based on the section and sort method selected, this option will send commands to the robotic arm to sort all the CDs in the selected section.

Sample Data:

Sample_CD_Archive_Data.txt:

```
ID;Title;Author;Section;X;Y;BarCode;Description;OnLoan
1;Busn History 2015;Management;B;1;57;43783278;Business History - tax records, events and achievements in 2015;No
2;Busn History 2016;Management;B;1;58;43783279;Business History - tax records, events and achievements in 2016;No
3;Busn History 2017;Management;B;1;59;43784280;Business History - tax records, events and achievements in 2017;No
4;Client A's Project;Dev Team;D;4;7;43784761;Client A's project files and copyright material.;No
5;Client A's Testing Records;Testing Team;T;2;18;43784880;Client A's project test plan, test cases and associated emails;No
6;Web Site 2017;Web Team;W;1;27;43784901;Our business' 2017 web site development and graphics files;No
7;HRM Records 2017;HR Team;B;1;60;43785102;Our business' 2017 human resource management records;No
8;Marketing Records 2017;Marketing;B;1;61;43785367;Our business' 2017 marketing and sales records;No
9;Client B's Project;Dev Team;D;4;15;43785444;Client B's project files and copyright material;No
10;Client A's Testing Records;Testing Team;T;2;32;43785445;Client B's project test plan, test cases and associated emails;No
11;Busn History 2018;Management;B;1;64;43785498;Business History - tax records, events and achievements in 2018;No
```

ID	Title	Author	Section	X	Y	BarCode	Description	OnLoan
1	Busn History 2015	Management	B	1	57	43783278	Business History - tax records, events and achievements in 2015	FALSE
2	Busn History 2016	Management	B	1	58	43783279	Business History - tax records, events and achievements in 2016	FALSE
3	Busn History 2017	Management	B	1	59	43784280	Business History - tax records, events and achievements in 2017	FALSE
4	Client A's Project	Dev Team	D	4	7	43784761	Client A's project files and copyright material.	FALSE
5	Client A's Testing Records	Testing Team	T	2	18	43784880	Client A's project test plan, test cases and associated emails	FALSE
6	Web Site 2017	Web Team	W	1	27	43784901	Our business' 2017 web site development and graphics files	FALSE
7	HRM Records 2017	HR Team	B	1	60	43785102	Our business' 2017 human resource management records	FALSE
8	Marketing Records 2017	Marketing	B	1	61	43785367	Our business' 2017 marketing and sales records	FALSE
9	Client B's Project	Dev Team	D	4	15	43785444	Client B's project files and copyright material	FALSE
10	Client A's Testing Records	Testing Team	T	2	32	43785445	Client B's project test plan, test cases and associated emails	FALSE
11	Busn History 2018	Management	B	1	64	43785498	Business History - tax records, events and achievements in 2018	FALSE

PART 1. Environment setup

1.1 Before you start the project, review the project requirements and documentation and articulate these into an email and in a meeting with your Project Manager. The purpose of the email and meeting is to confirm your understanding of the requirements and communicate your intention to start the project. Provide meeting minutes.

1.2 Review the current *CD Archive Prototype* records from the *CD_ArchivePrototype_SampleData.txt* data file.

1.3 Access the External Documentation template - in line with the development team's programming standards (*UptownIT_External_Documentation_Template.docx)*.

Use this template in documenting your design of the application, including:

a) IPO chart
b) TOE chart
c) DFD
d) Standard algorithms – using pseudocode
e) Class diagrams

PART 2. Build the application – code the solution

2.1 Select and use the most efficient user-defined data structures to code the functionality outlined in the *Scenario Script and Requirements* section on the previous pages. Ensure that you follow the algorithms created in 1.3 and adhere to the Uptown IT code standards.

2.2 In the development of your Java application, using an appropriate IDE, and as a Graphical User Interface application, ensure that you implement:

a) A complex data structure to store the sample client data that the program reads in.
b) A facility for the program operator to search for a record in this complex data structure.
c) A facility to display the data read in – both as a list and as an individual record.
d) Three (3) sort methods that visibly manipulate portions of the data read in. These are to be fully coded, not utilise pre-defined sort functionality available within the programming language or associated IDE.
e) A Doubly Linked List facility, as required by the specification and with appropriate search functionality.
f) A Binary Tree facility, as required by the specification and with appropriate search functionality
g) An output table demonstrating the application of hashing techniques.
h) Inter-process communication between the components of the system being developed, as outlined within the scenario, on the previous pages.
i) An applicable programmed response to an operating system signal.
j) The identification and utilisation of a relevant third-party library. Locate applicable documentation associated with the third-party library, to be noted and emailed to your project manager and client. (See item 4.5 on the following pages.)

k) Complete unit testing and utilise the IDE's debugging facilities in the ensuring the correct operation of your initial development code. Check for syntactical, logical and design errors.
l) Discuss implementation concerns and/or point of clarification with your project manager as the project proceeds. Make reference to these and the feedback provided in your closing email to your project manager.

PART 3. Program documentation

3.1 Create appropriate internal documentation in the form of in-line comments in the code. Your code should follow your development team's programming standards. Review the Code Standards document (UptownIT_Code_Standards.docx).

3.2 Access the external documentation template, UptownIT_External_Documentation_Template.docx and complete the following sections:
- Test Plan
- Test Cases Report (file: UptownIT_Test_Cases_Report_Template.docx)
- User Manual
- Evidence of correspondence (emails to your Project Manager and Client)

3.3 Review the debugging and test procedures outlined in the development team's coding standards (*UptownIT_Java_Coding_Standards_Template.docx).* Note down any adjustment you feel may be required to these standards (approaches) for this project or for future projects. Add these notes to your closing email to your manager.

3.4 Prepare automated program documentation using a facility provided by or made available within your IDE such as Javadoc. A third-party facility can also be used to prepare automated program documentation.

PART 4. Debugging, testing and optimising code

4.1 Provide evidence of your use of the debugging facilities available within the Integrated Development Environment (IDE) you are utilising. Check for syntactical, logical and design errors. This may be demonstrated with a series of screen images of you debugging your application. Showing at least:
a) one breakpoint
b) a set of associated watches
c) an instance of tracing through several lines of code

Add the screen images to the **external** documentation template, *UptownIT_External_Documentation_Template.docx.*

4.2 Carry out the Test Plan prepared in 3.2 and use the Test Cases Report (file: UptownIT_Test_Cases_Report_Template.docx) to document the results of the testing.

4.3 Research and apply at least two (2) code optimisation techniques. One might be optimisation by inspection, the other in response to feedback from a Java code analyser. Retain the detail identified for inclusion in your final email to your project manager. (See item 4.8.)

4.4 Update your application so as to resolve any errors, issues, optimisation adjustments, or oversights in relation to the original requirements specification that may have been noted during your testing. Pay attention not only to syntactical errors but also logical and design problems.

4.5 Prepare a supporting user manual appropriately considered and structured. Add the user manual to the **external** documentation template, *UptownIT_External_Documentation_Template.docx.*

4.6 Prepare **emails** to your Project Manager and separately for your client regarding the details of the **third party library** utilised within the application.

4.7 Prepare an **email** for your client providing feedback on the overall success of the project. Note any areas where the project has not been able to meet any of the specific project requirements, providing your associated reasoning and justification, and an outline of your alternate implementation(s).

4.8 Prepare an **email** to your Project Manager regarding the completion of the development, summarising your debugging and test results, and your code optimisation investigations. You should suggest any areas of **concern or risk** to your manager for inclusion in the development team's risk matrix. Include your notes in relation to any adjustment you may have made to the debugging and testing approaches you may have needed to use, that were otherwise not included in your team's coding standards.

4.9 Upload the code and the documentation, request feedback from your project manager, and look to obtain approval and signoff. Complete any follow up tasks based on feedback provided by your project manager, and resubmit your final solution plus explanatory notes if/as required.

PROGRAMMING PORTFOLIO - SIGNOFF Signing off on this document signifies that the submitted code and documentation comply with the Client Business requirements.	
Project Manager Signature: Date:	Programmer Signature: Date:
PROGRAMMING PORTFOLIO NOT APPROVED Please provide feedback on the changes needed.	

PART 5. Contingency and advanced programming concepts

5.1 **Contingency task**. You are the project manager for a new large-scale application project. Development is scheduled to start in 10 days and today the lead programmer assigned to the project has handed in his resignation. The organisation has two other programmers with similar expertise but they are fully utilised in other projects.
Outline a strategy to find a programmer (expert Java developer) to start the project.

5.2 A client project requires an undo and redo facility for a small and finite set of data entry sets that are added to and removed quite quickly. At a meeting with the project's systems analyst, you discuss what you believe would be the best-suited data structure for this facility.

After some discussion, you suggest to use a Doubly Linked List of Objects. Justify your recommendation.

5.3 A client project requires the storage of an object and its associated index value. The ability to efficiently locate a stored object is of highest priority in relation to the project's required functionality.

Propose the most appropriate data structure for the task and explain your selection.

5.4 Quick Sort is a very efficient sorting algorithm and is commonly utilised within program libraries for sorting sets of data.

Nevertheless, it is not always possible to utilise a Quick Sort from a program library, potentially because:

- The project's data set is being stored in a complex data structure not supported by the Quick Sort facility provided.
- A Quick Sort does not suit the required sort.

Outline four (4) conditions that the subsequent selection of a sort algorithm in a specific situation may depend on.

5.5 In creating an indexing system for a collection of data objects, a mathematical function (a hashing technique) could be utilised. The hashing technique or algorithm would be applied to the key of each data object providing for a set of individual numerical addresses. This allows for the efficient retrieval of a given data object once a search parameter has been provided, and has had the same hashing algorithm applied.

Identify and explain two (2) advantages of using a hashing technique.

5.6 One of the programming concepts that can be utilised in both inter-process communication and operating system signals is Threads.

Analyse the above statement and justify the use of Threads.

End of Assessment