RICHFIELD

FACULTY OF INFORMATION TECHNOLOGY

PROGRAMMING 731

1ST SEMESTER ASSIGNMENT

Name & Surname:____________________________ **ITS No:** ______________________________

Qualification:____________________**Semester:**_______**Module Name:** _____________________

Date Submitted: ________

ASSESSMENT CRITERIA	MARK ALLOCATION	EXAMINER MARKS	MODERATOR MARKS
MARKS FOR CONTENT			
QUESTION ONE	20		
QUESTION TWO	20		
QUESTION THREE	50		
TOTAL	90		
MARKS FOR TECHNICAL ASPECTS			
CODE LAYOUT/STRUCTURE AND COMMENTS	10		
TOTAL MARKS FOR ASSIGNMENT	100		
Examiner's Comments:			
Moderator's Comments:			
Signature of Examiner:	Signature of Moderator:		

ASSIGNMENT INSTRUCTIONS

1. All assignments must be typed, not handwritten.
2. Every assignment should include the cover page, table of contents and a reference list or bibliography at the end of the document.
3. A minimum of five current sources (references) should be used in all assignments, and these should be reflected in both in-text citations and the reference list or bibliography.
4. In-text citations and a reference list or bibliography must be provided. Use the Harvard Style for in-text citations and the reference list or bibliography.
5. Assignments submitted without citations and accompanying reference lists will be penalised.
6. Students are not allowed to share assignments with fellow students. Any shared assignments will attract stiff penalties.
7. Using and copying content from websites such as chegg.com, studocu.com, transtutors.com, sparknotes.com or any other assignment-assistance websites is strictly prohibited. This also applies to Wiki sites, blogs, and YouTube.
8. Any pictures and diagrams used in the Assignment should be labelled appropriately and referenced.
9. Correct formatting as indicated on the Cover Page should be followed (font size 12, font style Calibri, line spacing of 1.0 and margins justified)
10. All assignments must be saved in PDF using the correct naming convention before uploading them to Moodle: E.g., StudentNumber_CourseCode_Assignment (402999999_WBT512A_Assignment).

QUESTION 1 (20 MARKS)

Create a text file called *employee.txt.* The file should contain the following content:

Name	Surname	Years Worked	Salary
John	Smith	4	15000
Ayanda	Dube	10	200000
Damien	Naidoo	5	65000

Write a Java program that reads the contents of the text file. For each employee in the file, update their salary and overwrite the contents of the existing file to reflect the changes. Salary increase is based on the following criteria:

Years Worked	Increase
< 5	5%
5 - 10	15%
> 10	30%

NB: Make sure you catch all the exceptions that might occur.

QUESTION 2 (20 MARKS)

Design a Java program that implements a class named Task. This class should have three instance variables: taskName (a string), taskId (an integer), and taskWage (an integer). The class should have a constructor that initialises these variables.

In the main class, create two worker threads, each with its method to display the task details. The first method should display the task name, ID, and wage. The second method should display the same information but with some additional information about the task. Both methods should accept an object of type Task as a parameter.

The two worker threads should be paused for a specific amount of time (e.g. 1000ms) before they execute their methods. Additionally, use a relevant thread method to display the priority of each thread. Finally, check each thread's status and display each thread's name and status.

QUESTION 3 (50 MARKS)

Description of Blackjack:

Blackjack is a card game played against a casino dealer. The game's object is to collect cards in your hand whose total is as high as possible without going over 21. If the cards you hold beat the dealer's, you win!

How Blackjack is played:

The player and the dealer receive two cards. The player then counts the total points in their hand and decides to hit (take another card) or stand (accept no more cards). If you go over 21, you "bust" (lose). The player wins if they do not "bust" and their total points are higher than the dealer's.

Blackjack card values:

- Face cards count as 10.
- Aces count as either 1 or 11 - i.e., count as 11 as long the value does not exceed 21.
- All other cards count as their numeric value.

Player Actions:

- Hit Deals another card to the player's hand.
- Stand Indicates that the player does not want more cards and ends her turn.

Instructions:

- Create a Deck class that stores all cards in a deck in an ArrayList. This class also includes a shuffle and drawCard method.
- Create a Card class that will store and return the rank and suit of the cards.
- Create a Hand class to add the cards to the player's or dealer's hand. This class will include a method that calculates the points of the hand and a method to display the cards.
- Create a BlackjackGame class that simulates the game by creating methods to deal the cards to the player and the dealer, allowing players to hit or stand and determine and output the winner.