IVE Information Technology

Information & Communications Technology

Programme Board

HIGHER DIPLOMA IN

SOFTWARE ENGINEERING
(IT114105, IT314105)

MODULE TITLE:

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

MODULE CODE: **ITP4514**

SEMESTER 1
MAIN EXAMINATION

6 January, 2024
9:30 AM TO 11:30 AM (2 hours)

Instructions:

(a) This paper has a total of TWELVE pages including the covering page and appendices.

(b) This paper contains TWO Sections.

(c) Section A is worth 40 marks and Section B is worth 60 marks.

(d) Answer ALL questions in Section A. Each question is worth 8 marks.

(e) Answer ALL questions in Section B. Each question is worth 15 marks.

Note:The result of this assessment will not be counted if you do not meet the minimum attendance requirement (if any) governed by the general academic regulations of your programme/course unless approval of the campus principal has been granted.

[This Page is Intentionally Left Blank]

This paper contains TWO sections.

All working must be clearly shown on your answer book.

Section A (40 marks)

This section contains FIVE questions.
Answer ALL questions.
Each question is worth EIGHT marks.

A1 (a) Briefly explain **Traditional programming** and **AI**.

[2 marks]

(b) Write a Python program to print a ***triangle of permutations*** with a particular integer inputted by the user. For example, if the user's input is 5, the program should display:

```
Enter the size of triangle: 5

  1
  1  2
  1  2  3
  1  2  3  4
  1  2  3  4  5
```

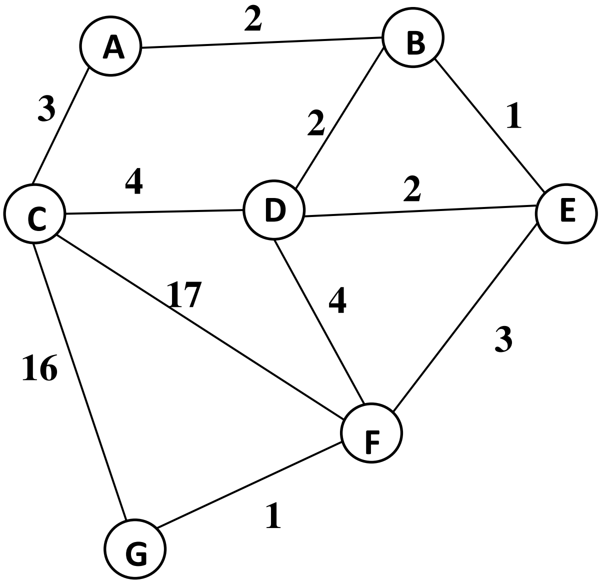
Note that each number on the triangle is computed by a function CalNum for each columns. For example, the second column is 2, the third column is 3, the fourth column is 4. The Python function CalNum is given as follow:

```
def CalNum(num):
        return num + 1
```

NOTE: You are required to include the code above in your answer. [4 marks]

(c) List **TWO** data types in Python. [2 marks]

A2 (a) Consider the following graph:



and the heuristic cost:

Node	H(n)
A	20
B	7
C	8
D	10
E	6
F	1
G	0

<u>Note</u>: The Start Node is A, and the Goal Node is G.

Explain how A* Search Algorithm would traverse the graph. [4 marks]

(Hint: You may use the following format as an initial step.)

Opened=[A20] **Closed=[]**

QUESTION A2 CONTINUES ON THE NEXT PAGE

QUESTION A2 CONTINUES FROM THE PREVIOUS PAGE

(b) You are given the following Python code:

```
from sklearn.datasets import load_wine
from sklearn.svm import SVC

wine = Part (b)(i): Add your code here
X = wine.data[: , 0:3]     # alcohol, maltic acid, ash contents
y = wine.target            # wine type: class-0, class-1 or class-2
……
Part (b)(ii): Add your code here
Part (b)(iii): Add your code here
```

Write Python code to complete the following tasks:

(i) Load and store the data in variable “wine”. [1 mark]

(ii) Create and train a Support Vector Machine model with the Wine dataset. [2 marks]

(iii) Predict the wine type if it is alcohol, maltic acid and ash content is 13, 1.5 and 2.5 respectively. [1 mark]

A3 (a) List **TWO** examples of uncertainty in *Probabilistic Reasoning*. [4 marks]

(b) Define the term “Constraints” in *Constraint Satisfaction Problems (CSP).* [4 marks]

A4 (a) List **FOUR** types of machine learning. [4 marks]

(b) Briefly explain the differences between **Regression** and **Classification**. [2 marks]

(c) List **ONE** example of output values in ***regression***, and **ONE** example of output values in ***classification***. [2 marks]

A5 (a) Write down **THREE** calculation methods which are commonly used to quantify similarity in recommendation systems. [3 marks]

(b) Write down any **THREE** layers which can be found in Convolutional Neural Network. [3 marks]

(c) What kind of activation function is commonly used in the final layer of a CNN classifier? [2 marks]

Section B (60 marks)

This section contains FOUR questions.
Answer ALL questions from this section.
Each question is worth FIFTEEN marks.

B1 (a) (i) In this question, you are required to create **2022 epidermic table** by utilizing *Pandas* library. The frame stores the last four months of the records. Write Python code for assigning the following DataFrame to the variable, ***frame***.

[4 marks]

	Level	Month	Cases
Sep	2	September	216
Oct	3	October	990
Nov	4	November	325
Dec	5	December	3583
UC	unclassified	Others	493

(ii) Insert the *Covid, Influenza, Monkeypot* columns for the table below. You are not allowed to create a new DataFrame other than using ***frame*** as defined in **B1(a)(i)**: [3 marks]

	Level	Month	Covid	Influenza	Monkeypot	Cases
Sep	2	September	100	110	6	216
Oct	3	October	888	100	2	990
Nov	4	November	234	88	3	325
Dec	5	December	1234	2345	4	3583
UC	unclassified	others	137	150	206	493

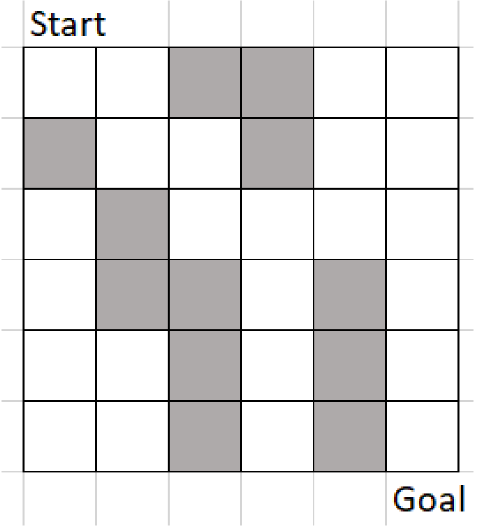
(iii) Write the Python code by removing the **"unclassified"** row.

[1 mark]

	Level	Month	Covid	Influenza	Monkeypot	Cases
Sep	2	September	100	110	6	216
Oct	3	October	888	100	2	990
Nov	4	November	234	88	3	325
Dec	5	December	1234	2345	4	3583

(iv) State **ONE** reason using Numpy instead of list to manipulate array in machine learning. [2 marks]

B1 (b) Consider the following Python code, which implements the ***Best First Search*** Algorithm on a 6*6 grid:



```
grid = [[0, 0, 1, 1, 0, 0],
        [1, 0, 0, 1, 0, 0],
        [0, 1, 0, 0, 0, 0],
        [0, 1, 1, 0, 1, 0],
        [0, 0, 1, 0, 1, 0],
        [0, 0, 1, 0, 1, 0]]

heuristic = [[13,12,11, 9, 7, 6],
             [12,11,10, 8, 6, 5],
             [11,10, 8, 7, 5, 4],
             [10, 9, 8, 6, 5, 2],
             [ 8, 8, 6, 5, 3, 1],
             [ 7, 5, 4, 2, 1, 0]]

init = [0, 0]
goal = [len(grid)-1, len(grid[0])-1]
cost = 0

delta = [[-1, 0 ], # go up
         [ 0, -1], # go left
         [ 1, 0 ], # go down
         [ 0, 1 ]] # go right

delta_name = ['^', '<', 'v', '>']
```

QUESTION B1 CONTINUES ON THE NEXT PAGE

QUESTION B1 CONTINUES FROM THE PREVIOUS PAGE

```
def search(grid,init,goal,cost,heuristic):
    # ----------------------------------------
    # modify the code below
    # ----------------------------------------
    closed = [[0 for col in range(len(grid[0]))] for row in range(len(grid))]
    closed[init[0]][init[1]] = 1

    expand = [[-1 for col in range(len(grid[0]))] for row in range(len(grid))]
    action = [[-1 for col in range(len(grid[0]))] for row in range(len(grid))]

    x = init[0]
    y = init[1]
    g = 0
    f = g + heuristic[x][y]

    open = [[f, g, x, y]]

    found = False  # flag that is set when search is complete
    resign = False # flag set if we cannot find expand
    count = 0

    while not found and not resign:
        if len(open) == 0:
            resign = True
            return "Fail"
        else:
            open.sort()
            open.reverse()
            next = open.pop()
            f = next[0]
            g = next[1]
            x = next[2]
            y = next[3]

            expand[x][y] = count
            count += 1

            if x == goal[0] and y == goal[1]:
                found = True
```

QUESTION B1 CONTINUES ON THE NEXT PAGE

QUESTION B1 CONTINUES FROM THE PREVIOUS PAGE

```
            else:
              for i in range(len(delta)):
                  x2 = x + delta[i][0]
                  y2 = y + delta[i][1]
                  if x2 >= 0 and x2 < len(grid) and y2 >=0 and y2 < len(grid[0]):
                      if closed[x2][y2] == 0 and grid[x2][y2] == 0:
                          g2 = g + 0
                          f = g2 + heuristic[x2][y2]
                          open.append([f, g2, x2, y2])
                          closed[x2][y2] = 1

    return expand

result = search(grid,init,goal,cost,heuristic)

for el in result:
  print (el)
```

(i) Write down the output. [3 marks]

(ii) Copy the grid and draw the path found. [2 marks]

B2 (a) Below is the Solution for an **N-Queen Problem** when N=5:

```
// import necessary library
(i)________________________________________

// new a problem
prob = Problem()

// add variables
(ii)________________________________________(cols, rows)

for col1 in cols:
    for col2 in cols:
        if (iii)__________________________________:

                               // Add Constraints
                               (iv)_____________________________(lambda
                               row1, row2, col1=col1, col2=col2:

                               // Constraint 1
               (v)______________________________ and
                               // Constraint 2
               (vi) ____________________, (col1, col2))

solutions = prob.getSolutions()
```

(i) Fill in the blanks from **(i)** to **(vi)**. [6 marks]

(ii) What are the **<u>TWO</u>** techniques of *Logic Programming Engine (LPE)*? [4 marks]

QUESTION B2 CONTINUES ON THE NEXT PAGE

QUESTION B2 CONTINUES FROM THE PREVIOUS PAGE

(b) You are given the following dataset on credit card transactions from a bank. Fault transactions were labeled as 1 in the "fault" attribute.

tx.csv *(the first few lines is shown here)*

tx_no	frequency	amount	fee	fault
1	3	4000	23	0
2	20	58000	40	1
3	2	34000	25	0
4	4	890000	670	0
5	10	1020000	1290	0
6	6	23000	200	1

To classify 'fault' transactions based on attributes 'frequency', 'amount' and 'fee', the following code fragment is given:

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
dd = pd.read_csv("/content/tx.csv", header = 0)
X = # Part (b) (i): Add your answer here
y =  # Part (b) (i): Add your answer here
# Splitting the dataset into training and test set.
_# Part (b) (ii): Add your answer here
 # Part (b) (ii): Add your answer here

# K-Nearest Neighbors Classifier model
knn = # Part (b) (iii): Add your answer here
# train the model using training set
# Part (b) (iii): Add your answer here
#Predicting the test set result
y_pred = knn. # Part (b) (iii): Add your answer here (X_test)
……
```

Write Python code with Scikit-Learn to complete the following tasks:

(i) Obtain X from the columns "frequency", "amount" and "fee"; and y from the last column. [1 mark]

(ii) Split X and y into training set and testing set with ratio **8:2** . [2 marks]

(iii) Train using K-Nearest Neighbors algorithm and predict y for X_test. [2 marks]

B3 (a) Given a dataset with points: A(2, 10), B(2, 5), C(8, 4), D(12, 11). Conduct a **hierarchical clustering (bottom-up)** and write down your steps. In each step, specify the number of clusters, points in each cluster, and centroid of each cluster.

[10 marks]

(b) You are given the following 3 documents:

Document 1: ITP4514 is an AI course in SE
Document 2: SE students learn AI, ML and DL
Document 3: AI includes ML and DL

(i) Calculate the Term Frequency of the 3 documents, focusing on only the ***four words***:

ITP4514, AI, SE, ML. [3 marks]

(ii) Calculate the **Inverse Document Frequency**. [2 marks]

B4 (a) The following questions are regarding recommender systems.

(i) ***Content-based filtering*** and ***collaborative filtering*** are two common candidate generation approaches. Define what ***collaborative filtering*** does. Give an example of content-based filtering in *video recommendation system*.

[3 marks]

(ii) Freshness, diversity, and fairness can help improve a recommendation system. Suggest **ONE** method to enhance the ***diversity*** of the system. [2 marks]

(b) Consider the following questions are regarding deep learning.

(i) Suggest **ONE** commonly used ***activation function*** in *artificial neural networks*.
[1 mark]

(ii) What is the purpose of a normalizer in *artificial neural networks*? [2 marks]

(c) The following questions are regarding CNN.

(i) Perform a ***convolution operation*** with the following image and filter. [5 marks]

(ii) Perform a ***max-pooling operation of window size = 2x2*** with the same image.
[2 marks]

Image

5	0	3	1
2	1	2	0
0	4	1	2
1	0	2	0

Filter

1	-1
0	1

***** **END OF PAPER** *****