COMP4005 Object Oriented Programming Portfolio Template

You may use this file as a template for the Turnitin submission of your Portfolio Exercises.

You should paste your solutions submitted in Replit in a fixed width font such as Courier New.

# Portfolio and Engagement Marks 60%

This is the first submission of your portfolio exercises: weeks 1-4 (**10 marks**)

This assessment component is worth 60% of the marks for the module:

* 10 marks for the first submission of the portfolio exercises, this submission
* 40 marks for the second submission in week 11
* 10 marks for engagement

# Academic Conduct

The contents of your portfolio must be your own, individual work. You should ensure that you are aware of the university rules on plagiarism and collusion. **You are not allowed to use any AI tool such as ChatGPT to complete these exercises.**

# Portfolio Exercises

Put your submissions to the portfolio exercises here. When pasting Python code into your portfolio, use a fixed width font such as Courier New so as to preserve indentation.

## Week 1 (2 marks)

###### WK1P1 Parcel

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    double w, h, l, s;

    Scanner kb = new Scanner(System.in);

    System.out.print("Enter the width of the parcel: ");

    w = kb.nextDouble();

    System.out.print("Enter the height of the parcel: ");

    h = kb.nextDouble();

    System.out.print("Enter the length of the parcel: ");

    l = kb.nextDouble();

    s = w+h+l;

    if (s>90 || w>60 || h>60 || l>60) {

      System.out.print("parcel cannot be sent\n");

    }

    else {

      System.out.print("parcel can be sent\n");

    }

  }

}

###### WK1P2 Jugs

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    int ca, cb, va, vb, t, tf;

    Scanner kb = new Scanner(System.in);

    ca = 5;

    cb = 3;

    va = 0;

    vb = 0;

    t = 4;

    while ((va != t) && (va+vb != t)) {

      if (va == 0) {

        va = ca;

        System.out.println("Fill jug A with " + va + " litres");

      }

      if (vb == cb) {

        vb = 0;

        System.out.println("Empty jug B");

      }

      else {

        if (va < cb-vb) {

          tf = va;

        }

        else {

          tf = cb-vb;

        }

        va -= tf;

        vb += tf;

        System.out.println("Transfer " + tf + " litres from jug A to jug B");

      System.out.println("There are now " + va + " litres in jug and " + vb + " litres in jug B");

      }

    }

    if (va == t) {

      System.out.println("Jug A contains the desired quantity of water");

    }

    else {

      System.out.println("Transfer " + vb + "to jug A");

      va += vb;

      vb = 0;

      System.out.println("Jug A contains the desired quantity of water " + va);

    }

  }

}

## Week 2 (2 marks)

###### WK2P1 Vowels

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    Scanner kb = new Scanner(System.in);

    String input, lower;

    int l, a, e, i, o, u;

    a = 0;

    e = 0;

    i = 0;

    o = 0;

    u = 0;

    System.out.println("Enter a string: ");

    input = kb.nextLine();

    lower = input.toLowerCase();

    l = lower.length();

    for (int j=0; j<l; j++) {

      char check = lower.charAt(j);

      switch(check) {

        case 'a':

          a++;

          break;

        case 'e':

          e++;

          break;

        case 'i':

          i++;

          break;

        case 'o':

          o++;

          break;

        case 'u':

          u++;

          break;

      }

    }

    System.out.println("A: " + a);

    System.out.println("E: " + e);

    System.out.println("I: " + i);

    System.out.println("O: " + o);

    System.out.println("U: " + u);

  }

}

###### WK2P2 Nought and Crosses

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    char[][] board = new char[3][3];

    for (int row = 0; row < board.length; row++) {

      for (int col = 0; col < board[row].length; col++) {

        board[row][col] = '.';

      }

    }

    char thisPlayer = 'X';

    while (!isBoardFull(board)) {

      displayBoard(board);

      System.out.println("Character to be placed is: " + thisPlayer);

      while (true) {

        System.out.print("Enter the row at which you wish to place it: ");

        int prow = scanner.nextInt();

        int pr = prow-1;

        System.out.print("Enter the column at which you wish to place it: ");

        int pcol = scanner.nextInt();

        int pc = pcol-1;

        if (pr >= 0 && pr < 3 && pc >= 0 && pc < 3 && board[pr][pc] == '.') {

        board[pr][pc] = thisPlayer;

        break;

        }

        else {

          displayBoard(board);

          System.out.println("Invalid input or space already taken. Try again");

        }

      }

      thisPlayer = (thisPlayer == 'X') ? 'O' : 'X';

    }

    displayBoard(board);

    System.out.println("The board is full. Game is done");

  }

  public static boolean isBoardFull(char[][] board) {

    for (char[] row : board) {

      for (char cell : row) {

        if (cell == '.') {

          return false;

        }

      }

    }

  return true;

  }

  public static void displayBoard(char[][] board) {

    System.out.println("Current Board:");

    for (char[] row : board) {

      for (char cell : row) {

        System.out.print(cell + " ");

      }

      System.out.println();

    }

  }

}

## Week 3 (2 marks)

###### WK3P1 Car

public class Car {

  private String regNo;

  private double mileage;

  private double fuelUsed;

  public Car(String regNo) {

*this*.regNo = regNo;

  }

  public String getRegNo() {

    return regNo;

  }

  public double getMileage() {

    return mileage;

  }

  public double getFuelUsed() {

    return fuelUsed;

  }

  public double getMPG() {

    if (mileage > 0) {

      double MPG = mileage/fuelUsed;

      return MPG;

    }

    else {

      return -1;

    }

  }

  public void setMileage(double mileage) {

*this*.mileage = mileage;

  }

  public void setFuelUsed(double fuelUsed) {

*this*.fuelUsed = fuelUsed;

  }

}

###### WK3P2 Array of Car

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    Car[] cars = makeCarArray();

    String finished = "N";

    while (!finished.equals("Y")) {

      System.out.print("Enter a car registration number: ");

      String regNo = scanner.nextLine();

      Car car = findCar(regNo,cars);

      if  (car != null) {

        printCar(car);

      }

      else {

        System.out.println("Car not found");

      }

      System.out.print("Finished [Y/N]? ");

      finished = scanner.nextLine();

    }

  }

  public static Car findCar(String regNo, Car[] cars) {

    for (Car car: cars) {

      if (regNo.equals(car.getRegNo())) {

        return car;

      }

    }

    return null;

  }

  public static void printCar(Car car) {

    System.out.println("Mileage = " + car.getMileage());

    System.out.println("Fuel Used = " + car.getFuelUsed());

    System.out.println("MPG = " + car.getMPG());

  }

  public static Car[] makeCarArray() {

    Car[] cars = new Car[3];

    cars[0] = new Car("QW56TYU");

    cars[0].setMileage(340.4);

    cars[0].setFuelUsed(5.1);

    cars[1] = new Car("TW65ZPQ");

    cars[1].setMileage(256.5);

    cars[1].setFuelUsed(4.2);

    cars[2] = new Car("KE71HGF");

    cars[2].setMileage(987.3);

    cars[2].setFuelUsed(8.4);

    return cars;

  }

}

## Week 4 (4 marks)

###### UML Diagram

###### **A screenshot of a graph AI-generated content may be incorrect.**

###### Feature Class

import java.util.Scanner;

public class Feature {

  private String name;

  private double x;

  private double y;

  public Feature(String name, double x, double y) {

*this*.name = name;

*this*.x = x;

*this*.y = y;

  }

  public String getName() {

    return name;

  }

  public double getX() {

    return x;

  }

  public double getY() {

    return y;

  }

}

###### Chart Class

import java.util.Scanner;

public class Chart {

  private String name;

  private Feature[] features;

  private int numFeatures;

  public Chart(String name) {

*this*.name = name;

    features = new Feature[10];

    numFeatures = 0;

  }

  public void addFeature(Feature feature) {

    features[numFeatures] = feature;

    numFeatures++;

  }

  public void deleteFeature(int pos) {

    int i = pos;

    while (i<numFeatures-1) {

      features[i] = features[i+1];

      i++;

      }

    numFeatures--;

  }

  public int getNumFeatures() {

    return numFeatures;

  }

  public Feature getFeature(int pos) {

    int j = pos;

    if (j >= 0 && j < numFeatures) {

      return features[j];

    }

    else {

      return null;

    }

  }

  public Feature getNearestFeature(double x, double y) {

    Feature nearestFeature = null;

    double min\_distance = 100;

    if (numFeatures!=0) {

      for (int i=0; i<numFeatures; i++) {

        Feature feature = features[i];

        double distance = getDistance(feature, x, y);

        if (distance<min\_distance) {

          nearestFeature = features[i];

          min\_distance = distance;

        }

      }

    return nearestFeature;

    }

    else {

      return nearestFeature;

    }

  }

  public double getDistance(Feature feature, double x, double y) {

    double dx = feature.getX() - x;

    double dy = feature.getY() - y;

    double distance = Math.sqrt(dx\*dx + dy\*dy);

    return distance;

  }

}