# Assessment Task 2: Build a

simple OOP application

|  |  |
| --- | --- |
| Course code and  title | **ICT50220 Diploma of Information Technology** |
| Unit code and  title | **ICTPRG549 Apply intermediate object-oriented language skills** |
| Due date | 28 / 05 / 2024 |
| Resources required | Computer Access with Microphone & Video (optional) & Internet Melbourne Polytechnic lab PC that connected to the internet |
| Decision making  rules | To achieve an overall satisfactory result for this assessment task:   * Learners must achieve a satisfactory result for each item in the Observation and/or Assessment Checklist/s |
| Learner instructions | You are required to create a simple object-oriented application. There are 2 parts to this task:  **Part A** – respond to a series of questions to create your application. **Part B** - demonstrate the functionality of your application to your Assessor.  For this task you will:   * Complete it individually. * Write answers to all questions. * Demonstrate your application to your Assessor. * Complete it in class as indicated by your Assessor and submit it by the due date. * Have time to read and review the assessment task in class. * Submit your answers electronically via Moodle, (or in hard copy to your assessor, including the signed cover sheet and   Learner declaration)   * You must answer electronically and save the document as Assessment Task 2 StudentID.docx * You must agree (by clicking on the ‘I confirm radio button) with the assessment submission terms and conditions in Melbourne Polytechnic Moodle prior to the submission   If you have any questions about the task or concerns about your ability to complete the task, please discuss this with your Assessor. |

### Task Details

For this task you will create a simple object-orientated program (OOP) application. You will:

* use different OOP techniques to create the application
* check the code optimisation.

Once complete you need to explain and demonstrate the full program functionality to your Assessor. Scenario

Melbourne Polytechnic (MP), formerly NMIT, is a [vocational education](https://en.wikipedia.org/wiki/Vocational_education) ([TAFE)](https://en.wikipedia.org/wiki/TAFE) and [higher](https://en.wikipedia.org/wiki/Higher_education) [education](https://en.wikipedia.org/wiki/Higher_education) institute located in [Melbourne.](https://en.wikipedia.org/wiki/Melbourne) It is predominantly in the northern suburbs with 7 campuses and 4 training sites across Victoria.

|  |  |
| --- | --- |
| **Campuses**  Preston Collingwood Epping Heidelberg Prahran  Greensborough | **Training Sites**  Broadmeadows Eden Park  Yan Yean Ararat |

MP needs to build a simple student car park application to activate a student’s parking session through one computer, to be deployed in all parking areas.

The system requirements are:

* An easy and interactive GUI, include a textbox, button and form.
* Data required: student name, Student ID and car registration number.
* No database required at this stage.
* Function, e.g. check email and callback.
* Sorting by student ID.
* Search by student ID.
* Contract duration: 3 weeks.

You must now build the application to these specifications and answer the following questions. You need to provide screenshots as evidence throughout the development of the application, ensure to review the following questions prior to commencing your application build.

Once complete contact your Assessor to explain and demonstrate the functionality of your application.

[YOU SHOULD USE YOUR INITIALS AT THE BEGINNING OF EACH VARIABLE](https://www.google.com/search?rlz=1C1CHBD_en-GBAU856AU856&sxsrf=ALeKk00q5GW_WoRCZ26o5t94Dls_ARuSxg%3A1621762887516&q=%E2%80%A2%2BYOU%2BSHOULD%2BUSE%2BYOUR%2BINITIALS%2BAT%2BTHE%2BBEGINNING%2BOF%2BEACH%2BVARIABLE&spell=1&sa=X&ved=2ahUKEwidvMfGwd_wAhVozDgGHV3lAdsQkeECKAB6BAgBEDE)

i.e. my name is Hatim Mansor, so variables should be: HM\_FirstName

HM\_LastName HM\_Counter HM\_i

### Part A - Development

Answer the following questions and develop the application to the specifications provided in the scenario.

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **Read the scenario above and determine the program requirements.** | | |
|  | ANSWER |  SATISFACTORY |  NOT SATISFACTORY |
| **PC specification**  **Computer name:XIAOMIAN**  **Operating System: Windows 11 Professional 64-bit**  **BIOS: G513QM.331**  **Processor: AMD Ryzen 9 5900HX with Radeon Graphics**  **Memory: 16384MB RAM**  **Graphics Card: NVIDIA GeForce RTX3060 Laptop GPU** | | | |
| **OOP Data Structure**  **List** | | | |
| **Main data Fields**  **student name, Student ID and car registration number** | | | |
| **Duration**  **3 Weeks** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2** | **Based on the scenario, select 2 functions to incorporate into your application. Provide a screenshot for each function..**  **Take a screenshot for two of the functions that you have used in your project.** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
| **Function 1:** | | | |
| **Function 2:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **3** | **Create the code for data sorting and data searching facilities in your application.**  **Supply a screen shot of each code below.** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
| **Data Sorting Code:** | | | |
| **Searching facilities code:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **4** | **Provide a screenshot for your main interactive GUI user interface** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT  SATISFACTORY |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **5** | **Provide a screenshot of the code you have created for the standard GUI components, textbox, button and form.** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
| **Textbox and Form:** | | | |
| **Button:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **6** | **Provide an example of user or program-generated error events you encountered during the application development. Include a screenshot of the error that appeared and explain how you resolved it.** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
| Error: Student number format is wrong  Solution: Fill in the student number according to the rules | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **7** | **Provide a screenshot of variable tracing whilst your program is running.** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **8** | **During your program development you would have faced some logical and coding errors, provide two screenshots**   1. **The system message** 2. **The logical and coding error** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
| **System Message:** | | | |
| **Coding Error:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **9** | **Provide a screenshot demonstrating that you have checked your variable content and any errors have been fixed.** | | |
|  | SCREENSHOT |  SATISFACTORY |  NOT SATISFACTORY |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **10** | **Copy and paste your written program here. Ensure your program has maintained program transactional integrity.** | | |
|  | ANSWER |  SATISFACTORY |  NOT SATISFACTORY |
| # Import the required libraries  import re  import tkinter  import tkinter.messagebox  from operator import attrgetter  # Student ID detection  def check\_student\_id(wyz\_stud\_id):      if re.match("^s[0-9]{7}$", wyz\_stud\_id, re.ASCII):          return True      else:          return False  # Student category  class Student:      def \_\_init\_\_(self, wyz\_name, wyz\_stud\_id):          try:              self.wyz\_name = wyz\_name              assert check\_student\_id(wyz\_stud\_id)              self.wyz\_stud\_id = wyz\_stud\_id          except AssertionError:              tkinter.messagebox.showerror("Error", "Student number format is wrong")              raise      # Locomotive number verification  def check\_rego\_number(wyz\_rego\_number):      if re.match("^c[0-9]{7}$", wyz\_rego\_number, re.ASCII):          return True      else:          return False  # Vehicle category  class Car:      def \_\_init\_\_(self, wyz\_rego\_number, wyz\_stud\_object):          try:              assert check\_rego\_number(wyz\_rego\_number)              self.wyz\_rego\_number = wyz\_rego\_number              self.wyz\_owner = wyz\_stud\_object          except AssertionError:              tkinter.messagebox.showerror("Error", "The vehicle registration number is incorrect")              raise  # Student Management  class StudentManager:      def \_\_init\_\_(self):          self.wyz\_stud\_list = []      def stud\_register(self, wyz\_stud\_object):          for s in self.wyz\_stud\_list:              if s.wyz\_stud\_id == wyz\_stud\_object.wyz\_stud\_id:                  tkinter.messagebox.showerror("Error", "Student already exists")                  return False          else:              self.wyz\_stud\_list.append(wyz\_stud\_object)              return True  # Vehicle management  class CarManager:      def \_\_init\_\_(self):          self.wyz\_car\_list = []      def c\_register(self, wyz\_car\_object):          for car in self.wyz\_car\_list:              if car.wyz\_rego\_number == wyz\_car\_object.wyz\_rego\_number:                  tkinter.messagebox.showerror("Error", "The vehicle has been registered!")                  return False          else:              self.wyz\_car\_list.append(wyz\_car\_object)              return True  # Search by student ID  def search\_studByID(wyz\_stud\_list, wyz\_stud\_id):      for s in wyz\_stud\_list:          match = re.match(wyz\_stud\_id.strip(), s.wyz\_stud\_id.strip())          if match:              return s      else:          return None      # Sort by student ID  def sort\_stdByID(wyz\_stud\_list, reverse=False):      wyz\_stud\_list.sort(key=attrgetter('wyz\_stud\_id'), reverse=reverse)      return wyz\_stud\_list      # Student and vehicle management class instantiation  wyz\_student\_manager = StudentManager()  wyz\_c\_manager = CarManager()  wyz\_s1 = Student("wyz\_1", "s6371683")  wyz\_student\_manager.stud\_register(wyz\_s1)  wyz\_s1\_car1 = Car("c6243582", wyz\_s1)  wyz\_c\_manager.c\_register(wyz\_s1\_car1)  wyz\_s2 = Student("wyz\_2", "s8496131")  wyz\_student\_manager.stud\_register(wyz\_s2)  wyz\_s2\_car1 = Car("c8002805", wyz\_s2)  wyz\_c\_manager.c\_register(wyz\_s2\_car1)  wyz\_s2\_car2 = Car("c2223457", wyz\_s2)  wyz\_c\_manager.c\_register(wyz\_s2\_car2)  wyz\_s3 = Student("wyz\_3", "s1153706")  wyz\_student\_manager.stud\_register(wyz\_s3)  wyz\_s3\_car1 = Car("c5892447", wyz\_s3)  wyz\_c\_manager.c\_register(wyz\_s3\_car1)  wyz\_s3\_car2 = Car("c6509401", wyz\_s3)  wyz\_c\_manager.c\_register(wyz\_s3\_car2)  wyz\_s3\_car3 = Car("c2515416", wyz\_s3)  wyz\_c\_manager.c\_register(wyz\_s3\_car3)  wyz\_s4 = Student("wyz\_4", "s1638122")  wyz\_student\_manager.stud\_register(wyz\_s4)  wyz\_s4\_car1 = Car("c3402583", wyz\_s4)  wyz\_c\_manager.c\_register(wyz\_s4\_car1)  wyz\_s4\_car2 = Car("c6677834", wyz\_s4)  wyz\_c\_manager.c\_register(wyz\_s4\_car2)  wyz\_s5 = Student("wyz\_5", "s9326799")  wyz\_student\_manager.stud\_register(wyz\_s5)  wyz\_s5\_car1 = Car("c8988361", wyz\_s5)  wyz\_c\_manager.c\_register(wyz\_s5\_car1)  wyz\_s5\_car2 = Car("c8461502", wyz\_s5)  wyz\_c\_manager.c\_register(wyz\_s5\_car2)  wyz\_s6 = Student("wyz\_6", "s8937458")  wyz\_student\_manager.stud\_register(wyz\_s6)  wyz\_s6\_car1 = Car("c3156668", wyz\_s6)  wyz\_c\_manager.c\_register(wyz\_s6\_car1)  wyz\_s7 = Student("wyz\_7", "s6370163")  wyz\_student\_manager.stud\_register(wyz\_s7)  wyz\_s7\_car1 = Car("c4449979", wyz\_s7)  wyz\_c\_manager.c\_register(wyz\_s7\_car1)  wyz\_s7\_car2 = Car("c3108408", wyz\_s7)  wyz\_c\_manager.c\_register(wyz\_s7\_car2)  wyz\_s7\_car3 = Car("c9003319", wyz\_s7)  wyz\_c\_manager.c\_register(wyz\_s7\_car3)  wyz\_s8 = Student("wyz\_8", "s1404312")  wyz\_student\_manager.stud\_register(wyz\_s8)  wyz\_s8\_car1 = Car("c1148342", wyz\_s8)  wyz\_c\_manager.c\_register(wyz\_s8\_car1)  wyz\_s8\_car2 = Car("c2049844", wyz\_s8)  wyz\_c\_manager.c\_register(wyz\_s8\_car2)  wyz\_s9 = Student("wyz\_9", "s6886971")  wyz\_student\_manager.stud\_register(wyz\_s9)  wyz\_s9\_car1 = Car("c7897652", wyz\_s9)  wyz\_c\_manager.c\_register(wyz\_s9\_car1)  wyz\_s10 = Student("wyz\_10", "s7538364")  wyz\_student\_manager.stud\_register(wyz\_s10)  wyz\_s10\_car1 = Car("c5605246", wyz\_s10)  wyz\_c\_manager.c\_register(wyz\_s10\_car1)  wyz\_s10\_car2 = Car("c4588459", wyz\_s10)  wyz\_c\_manager.c\_register(wyz\_s10\_car2)  # Registered student information sorted by student ID  print(sort\_stdByID(wyz\_student\_manager.wyz\_stud\_list))  # Register to respond to events  def register():      wyz\_name = wyz\_e1.get()      wyz\_id = wyz\_e2.get()      wyz\_rego\_number = wyz\_e3.get()        if wyz\_name == '' or wyz\_id == '' or wyz\_rego\_number == '':          tkinter.messagebox.showerror("Error", "All information cannot be empty")          return False        wyz\_find = search\_studByID(wyz\_student\_manager.wyz\_stud\_list, wyz\_id)      if wyz\_find and wyz\_find.wyz\_name == wyz\_name:          try:              wyz\_new\_car = Car(wyz\_rego\_number, wyz\_find)          except:              return False            if wyz\_c\_manager.c\_register(wyz\_new\_car):              tkinter.messagebox.showinfo("Registration successful", "Vehicle registration successful")              return True          else:              return False      elif wyz\_find and wyz\_find.wyz\_name != wyz\_name:          tkinter.messagebox.showerror("Error", "Student number does not match student name")          return False      else:          try:              wyz\_new\_student = Student(wyz\_name, wyz\_id)          except:              return False            if wyz\_student\_manager.stud\_register(wyz\_new\_student):              tkinter.messagebox.showinfo("Registration successful", "Student registration successful")          else:              return False            try:              wyz\_new\_car = Car(wyz\_rego\_number, wyz\_new\_student)          except:              return False            if wyz\_c\_manager.c\_register(wyz\_new\_car):              tkinter.messagebox.showinfo("Registration successful", "Vehicle registration successful")              return True          else:              return False      # GUI graphical interface design  wyz\_root = tkinter.Tk()  wyz\_root.title("wyz\_Car\_Registration")  wyz\_root.resizable(width=False, height=False)  wyz\_lb1 = tkinter.Label(wyz\_root, text="wyz\_Car\_Registration", bg="darkorchid", fg="cornflowerblue", font=("Bold", 30))  wyz\_lb1.grid(row=0, column=0, columnspan=4, pady=10, ipadx=10, ipady=10)  wyz\_lb2 = tkinter.Label(wyz\_root, text="wyz\_Student Name", bg="lightsalmon", fg="aquamarine", font=("Bold", 15))  wyz\_lb2.grid(row=1, column=1, padx=10, pady=10, ipadx=5, ipady=5)  wyz\_e1 = tkinter.Entry(wyz\_root, font=("normal", 13))  wyz\_e1.grid(row=1, column=2, pady=10, ipadx=5, ipady=5)  wyz\_lb3 = tkinter.Label(wyz\_root, text="wyz\_Student ID", bg="lightsalmon", fg="aquamarine", font=("Bold", 15))  wyz\_lb3.grid(row=2, column=1, padx=10, pady=10, ipadx=5, ipady=5)  wyz\_e2 = tkinter.Entry(wyz\_root, font=("normal", 13))  wyz\_e2.grid(row=2, column=2, pady=10, ipadx=5, ipady=5)  wyz\_lb4 = tkinter.Label(wyz\_root, text="wyz\_Rego Number", bg="lightsalmon", fg="aquamarine", font=("Bold", 15))  wyz\_lb4.grid(row=3, column=1, padx=10, pady=10, ipadx=5, ipady=5)  wyz\_e3 = tkinter.Entry(wyz\_root, font=("normal", 13))  wyz\_e3.grid(row=3, column=2, pady=10, ipadx=5, ipady=5)  wyz\_btn = tkinter.Button(wyz\_root, text="wyz\_Register", command=register, bg="darkred", fg="paleturquoise", font=("normal", 15))  wyz\_btn.grid(row=4, column=1, columnspan=2, pady=10, ipadx=5, ipady=5)  wyz\_root.mainloop() | | | |

### Part B – Demonstration

You must now explain and demonstrate your application to your Assessor. Use your answers to the questions in Part A to assist you.

You will be assessed on the items in the observation checklist. Ensure to:

* be prepared to demonstrate and explain your application
* communicate appropriately and professionally
* explain the requirements of the application
* explain the functions you used
* demonstrate the sorting and data searching facilities
* demonstrate the GUI interface and components
* explain an error event, a logical error and a coding error and how you resolved this
* demonstrate the use of the applications variable tracing
* Explain how you maintained the transactional integrity.

## Observation Checklist 1: Task 2 - Build a simple OOP application

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Learner name** | | WangYiZhuo | | **Student ID** | | S1554654 | |
| **Assessor name** | | Qiao Yu | | **Date** | | 28/05/2024 | |
| OBSERVATION CHECKLIST  ASSESSOR TO COMPLETE THE FOLLOWING | | | | | | | |
| **THE LEARNER:** | | | | | **SATISFACTORY** | | **NOT SATISFACTORY** |
| 1 | Correctly explained the requirements of the application based on the scenario details. | | | | ☑ | | ☐ |
| 2 | Correctly explained and demonstrated applications use of functions. | | the | | ☑ | | ☐ |
| 3 | Correctly explained and demonstrated the applications use of sorting and data searching facilities. | | | | ☑ | | ☐ |
| 4 | Correctly explained and demonstrated applications interactive GUI interface. | | the | | ☑ | | ☐ |
| 5 | Correctly explained and demonstrated the applications use of GUI components textbox, button and form. | | | | ☑ | | ☐ |
| 6 | Provided an example of an error they encountered and how they resolved it. | | | | ☑ | | ☐ |
| 7 | Correctly explained and demonstrated applications use of variable tracing. | | the | | ☑ | | ☐ |
| 8 | Provided an example of a logical and coding error they encountered and how they resolved it. | | | | ☑ | | ☐ |
| 9 | Explained how they checked variable content and proved all errors were fixed. | | | | ☑ | | ☐ |
| 10 | Explained how they maintained program transactional integrity. | | | | ☑ | | ☐ |
| **Feedback -** Assessor must include feedback. | | | | | | | |
| The candidate has demonstrated all the aspect required above and basic satisfied the performance. | | | | | | | |

## Assessment Checklist: Task 2 - Build a simple OOP application

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Learner name** | | WangYiZhuo | **Student ID** | | S1554654 | |
| **Assessor name** | | Qiao Yu | **Date** | | 28/05/2024 | |
| ASSESSMENT CHECKLIST  ASSESSOR TO COMPLETE THE FOLLOWING | | | | | | |
| **THE LEARNER:** | | | | **SATISFACTORY** | | **NOT SATISFACTORY** |
| 1 | Correctly completed all questions in Part A | | | ☑ | | ☐ |
| 2 | Explained and demonstrated all items in Observation Checklist 1 | | | ☑ | | ☐ |
| **Feedback -** Assessor must include feedback. | | | | | | |
| The Candidate has correctly answered all questions in Part A and demonstrate all skills required in observation checklist. | | | | | | |

## Assessment Task Summary: Task 2: Build a simple OOP application

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRAINER/ASSESSOR TO COMPLETE THE FOLLOWING:  **THE LEARNER:** | | | | | | YES | NO |
| 1 | Satisfactorily answered all the questions | | | | | ☐ | ☐ |
| 2 | Satisfactorily completed all items in Observation Checklist 1 | | | | | ☐ | ☐ |
| FEEDBACK **-** Assessor must include feedback | | | | | | | |
|  | | | | | | | |
| OVERALL TASK RESULT | | | | | | | |
| * Satisfactory * Not Satisfactory (resubmission required) – Due date: | | | | | | | |
| DATE ASSESSMENT RETURNED | | |  | | | | |
| TRAINER/ASSESSOR NAME | | |  | | | | |
| TRAINER/ASSESSOR SIGNATURE | | | X | | | | |
| **LEARNER DECLARATION**: Please read, tick and sign below | | | | | | | |
| * I, WangYiZhuo have been advised of the outcome of this assessment task.   PRINT NAME | | | | | | | |
| LEARNER SIGNATURE | | WangYiZhuo | | DATE | 2024.04.28 | | |