



ICT 162
Object Oriented Programming

Tutor-Marked Assignment

July 2019 Presentation

TUTOR-MARKED ASSIGNMENT (TMA)

This assignment is worth 18% of the final mark for [ICT162, Object Oriented Programming](#).

The cut-off date for this assignment is **Wednesday, 16th October 2019, 2355 hours**.

Note to Students:

Submit your solution document in the form of a single MS Word file. You are to include the following particulars in your submission: Course Code, Title of the TMA, SUSS PI No., Your Name, and Submission Date. Put this information in the first page of your solution document. Use the template word document provided - **SUSS_PI-No_FullName_TMA.docx**. Rename the file with your suss PI and full name join with “_TMA” e.g. “**SUSS_PI-TomTanKinMeng_TMA.docx**” (without the quotes). Do NOT submit as a pdf document.

You should make only one submission for TMA.

You are to copy and paste Python source code into your solution document as text. If you submit source code as image, no marks will be awarded to your program. Submit screenshots for **only** output of your program, where applicable.

Question 1

Use object-oriented programming to write an application for a company that uses a subscription-based business model to loan out clothes such as office and party wear to female clients.

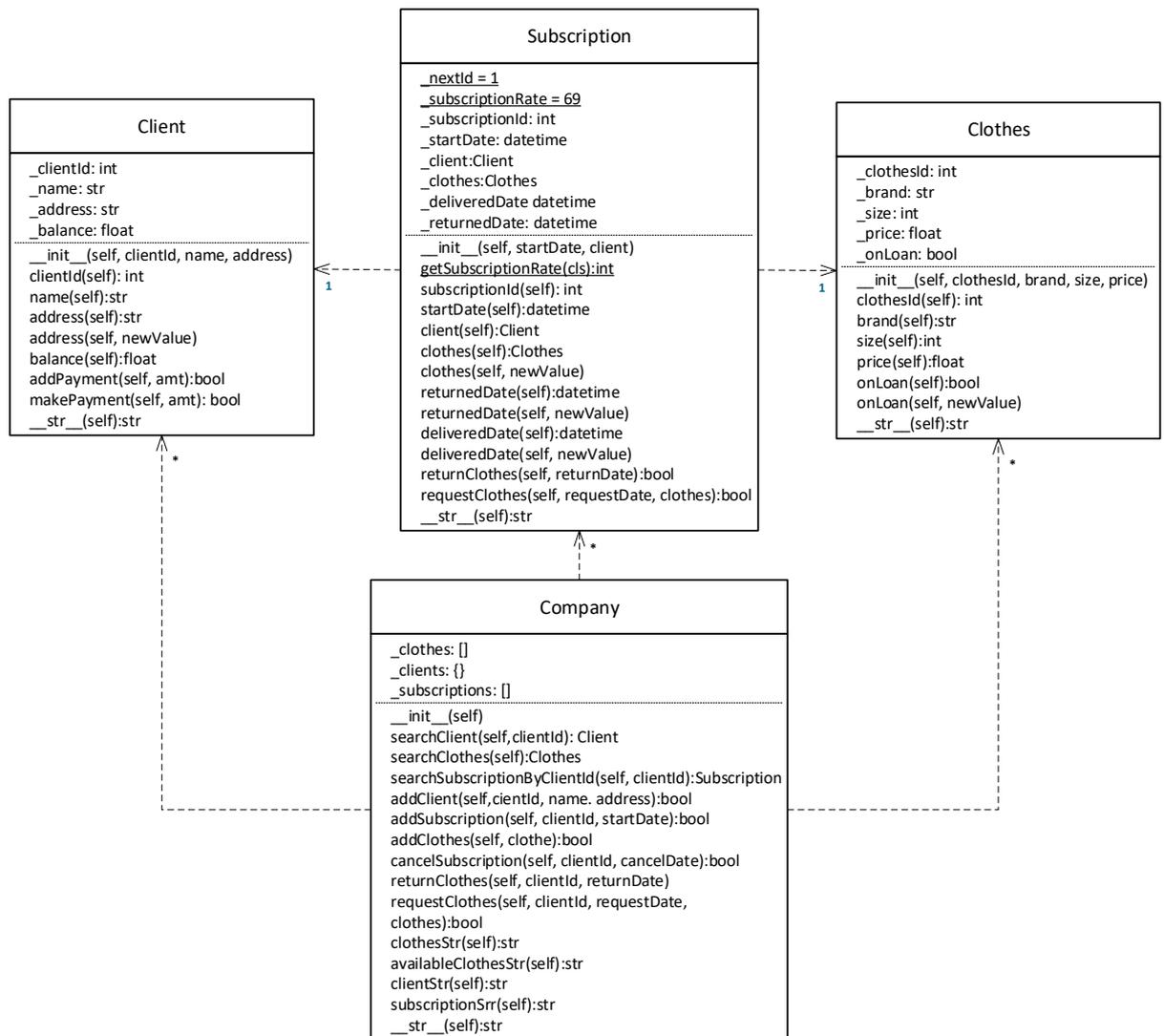
A subscription costs \$69 per 4 weeks. A client can make unlimited loan requests during her subscription, as long as she has no previous loan that has not been returned. There can be only one piece of clothes in each loan request.

The chosen piece of clothes takes a day to be delivered. For example, if a piece of clothes is selected on Wednesday (say, 4th September 2019), then it will arrive at the address of the client the following day, that is, Thursday, 5th 2019

The subscription is automatically renewed. To cancel subscription, a client must

- not have any loan that is unreturned,
- pay up any outstanding balance, and
- call to cancel before week 4 of each 4-week cycle.

The UML diagram for the application is shown in Figure Q1.

**Figure Q1**

Write the following classes:

(a) **Client**

- The object attributes are client id, name and address. The constructor has three parameters to initialise the three object attributes. There are accessor methods for all object attributes, and a mutator method for the address.
- A addPayment method has a parameter amount which is added to the client's balance if the amount is positive, and the method returns True. Otherwise, the method returns False.
- A makePayment method that has a parameter amount which is subtracted from balance if the amount is positive, and the method returns True. Otherwise, the method returns False. Note that the balance may become negative, and it means that the client has paid in excess.

- A `str` method that returns the object attributes as a string in the format shown here:

```
Id: 123 Name: Janet Teo Address: 21 Tampines West Road Payment due: $0
```

Test the class by creating the client in the example, and then invoke the methods you have written on the client object.

(2 marks)

(b) **Clothes**

- The object attributes are clothes id, brand, size, price and loan status. The loan status records True if the piece of clothes is currently loaned out, and False otherwise.

The constructor has four parameters to initialise the first four attributes. Set loan status to False.

There are accessor methods for each object attribute, and a mutator method for loan status.

- The `str` method returns all the values of the attributes as a string, with yes or no based on the loan status. Two examples are shown below:

```
Id: 104 Brand: Juillet Size: 6 Price: $214 Out: no
Id: 107 Brand: Elliatt Size: 7 Price: $237 Out: yes
```

Test the class by creating the two Clothes objects in the example, and then invoke the methods you have written on the Clothes objects.

(3 marks)

(c) **Subscription**

- The object attributes are the subscription id, start date of subscription, client who made the subscription, the piece of clothes loaned under the subscription, if any, the date on which the piece of clothes is delivered to the client if there is a loan, and the date on which the piece of clothes is returned, if the loan has been returned.

The constructor has two parameters to initialise the client and the start date of subscription. The subscription id is generated using a running number, implemented with a class attribute, `_nextId`. The subscription id of the first subscription is 1. The subscription id of the second subscription is 2, and so on. The piece of clothes, delivered date and return date are initialised to None.

There are accessor methods for each object attribute, and mutator methods for the piece of clothes, delivery date and return date.

- Another class attribute records the subscription rate which is \$69. A class method `getSubscriptionRate` returns the subscription rate.
- A `returnClothes` method has one parameter, return date, and method returns False if
 - there is no outstanding piece of clothes for the subscription or
 - the return date is before the date that the piece of clothes is delivered.
 Otherwise, the method handles the return of the piece of loaned clothes by setting the loan status of the piece of clothes to False, the object attribute

clothes to None, and updates the return date in the subscription. The method then returns True.

- A `requestClothes` method has two parameters, the request date and the piece of clothes. The method returns False if
 - a loan by the client has not been returned, or
 - the requested piece of clothes is currently already loaned out, or
 - the request date is before the start date of the subscription, or
 - the request date is before the returned date of the last loan.

Otherwise, the method handles the request to loan a piece of clothes by setting the delivered date for the piece of clothes to be one day after the request date, the loan status of the requested piece of clothes to True, the object attribute clothes to the requested piece of clothes, and returns True.

- The `str` method returns the object attributes as a string. Three examples are shown below:

Example 1: With no previous loan and no existing loan

Subscription Id: 1 Start Date: Sat, 31 Aug 2019
 Id: 123 Name: Janet Teo Address: 21 Tampines West Road Payment due: \$69
 No outstanding clothes.

Example 2: With a previous piece of clothes that is loaned, returned and no existing loan

Subscription Id: 1 Start Date: Sat, 31 Aug 2019
 Id: 123 Name: Janet Teo Address: 21 Tampines West Road Payment due: \$69
 No outstanding clothes. Returned: Fri, 06 Sep 2019

Example 3: With a loan of a piece of clothes

Subscription Id: 1 Start Date: Sat, 31 Aug 2019
 Id: 123 Name: Janet Teo Address: 21 Tampines West Road Payment due: \$69
 Clothes loaned Id: 107 Brand: Elliatt Size: 7 Price: \$237 Out: yes Delivered: Sun, 08 Sep 2019

Test the class by creating the Subscription object in the example, and then invoke the methods you have written on the Subscription object.

(10 marks)

(d) **Company**

- The object attributes are a collection of clothes (list), a collection of clients (dict) and a collection of subscriptions (list). The constructor initialises the three attributes as empty collections.
- There are three search methods, one for each of the object attributes.
 - A `searchClient` method returns a client in the collection that has the same client id as the parameter
 - A `searchClothes` method returns a piece of clothes in the collection that has the same clothes id as the parameter.
 - A `searchSubscription` method returns a subscription in the collection that is made by the client with the same client id as the parameter. Note that each client can make only one subscription.

The method returns None if the search is unsuccessful.

- There are three add methods, one for each of the object attributes
 - An `addClient` method that creates and adds a client to the collection only if the collection does not contain a client with the same client id.
 - An `addClothes` method that adds a piece of clothes to the collection only if the collection does not contain a piece of clothes with the same clothes id.
 - The `addSubscription` method checks that the client making the subscription is an existing client, and that the client does not have a subscription yet. If so, the method creates a subscription and adds it to the collection only if the collection does not contain a subscription with the same subscription id. The subscription amount is added to the payment of the client making the subscription.
 - The method returns `True` if the add operation is successful, and `False` otherwise.
- A `cancelSubscription` method that has two parameters, a client id and the cancel date. The method returns `False` if
 - there is no client with the given client id or
 - if the client has an outstanding balance, or
 - the client does not have a subscription, or
 - the cancel date is earlier than the start date of the subscription
 - the cancel date is not before week 4 of the 4-week subscription cycle,Otherwise, the method removes the subscription and returns `True`.
- A `returnClothes` method that has two parameters, a client id and the return date. The method returns `False` if there is no client with the matching client id or if the client does not have a subscription. Otherwise, the method calls the `returnClothes` method on the subscription, and returns the result of the call.
- A `requestClothes` method that has three parameters, a client id, the request date and the requested piece of clothes. The method returns `False` if
 - the requested piece of clothes is already on loan or
 - there is no client with the matching client id or
 - the client does not have a subscription.Otherwise, the method calls the `requestClothes` method on the subscription, and returns the result of the call.
- There are four `str` methods for object attributes.
 - A `clientStr` method returns a string with each client detail on separate lines.
 - A `clothesStr` method returns a string with each clothes detail on separate lines.
 - A `availableClothesStr` method returns a string with each clothes detail on separate lines, for clothes currently not on loan.
 - A `subscriptionStr` method returns a string with each subscription detail on separate lines.
- The `str` method returns the object attributes as a string in the format shown below:

Clients:

Id: 123 Name: Janet Teo Address: 21 Tampines West Road Payment due: \$69

Clothes

Id: 34 Brand: Charm Size: 7 Price: \$147 Out: no

Id: 91 Brand: Elliatt Size: 6 Price: \$287 Out: no

Id: 104 Brand: Juillet Size: 6 Price: \$214 Out: no

Id: 107 Brand: Elliatt Size: 7 Price: \$237 Out: yes

Id: 115 Brand: Juillet Size: 6 Price: \$253 Out: no

Id: 123 Brand: Suncoo Size: 6 Price: \$194 Out: no

Subscriptions

Subscription Id: 1 Start Date: Sat, 31 Aug 2019

Id: 123 Name: Janet Teo Address: 21 Tampines West Road Payment due: \$69

Clothes loaned Id: 107 Brand: Elliatt Size: 7 Price: \$237 Out: yes Delivered: Sun, 08 Sep 2019

Test the class by creating the Company object in the example, and then invoke the methods you have written on the Company object.

(10 marks)

Question 2

To enable collaborative learning through participation and engagement, you are encouraged to make postings in a discussion forum. You do not need to write Python code for this question as Python code is OPTIONAL.

The problem

Consider the same application described in question 1 where the company loans out clothes to clients on subscription. Suppose the company now wishes to expand its services to male clients as well as infrequent users, and clients who wish to buy a fixed duration subscription as a gift. A fixed duration subscription is not automatically renewed. Clothes that are not returned after the subscription ends will result in a fine on the client who purchased the gift subscription. .

- To cater to male clients, each piece of clothes now must indicate the gender it is applicable to.
- Besides the subscription with unlimited loan, the company will introduce two new subscription schemes to cater to infrequent users. The two schemes limit the number of loans to
 - 4 per 4-week at a subscription rate of \$49, and each loan is for one piece of clothes
 - 2 per 4-week at a subscription rate of \$39 with a maximum of 2 pieces per loan but restricted to a maximum of 3 pieces for the combined total of the 2 loans. Therefore, if the two loans are made with one loan for 2 pieces, then the next loan can be for only one piece of clothes.
- Gift subscription has a fixed duration - either 1, 3 or 6 cycles. Each cycle is 4 weeks.

The recipient details (name, address and age) must be recorded as well as the buyer, that is, the client. The company intends to provide a one-off discount to the gift recipients as enticement to subscribe two weeks before their gift expires.

As there are times when a piece of clothes is damaged or lost, the company intends to make the penalty structure transparent to all its clients. The penalty is dependent on the number of times the piece of clothes has already been loaned out as well as the number of 4-week cycles the subscription has undergone, according to Table Q2.

Damaged clothes loan count	< 3	< 8	< 15	>= 15
Length of subscription (#4-week)	Penalty (% of clothes price)			
< 6	100	90	75	60
<12	90	80	60	50
>= 12	80	70	50	30

Table Q2

You are to re-design the classes in Figure Q1 of Question 1, and introduce any other classes if necessary, to clearly demonstrate your understanding of inheritance and object composition, abstract classes and abstract methods, instance (or object) variables and class variables, instance (or object) methods and class methods and overriding methods.

You do not need to write Python code for this question; Python code is OPTIONAL.

(a) Learning Journey

This is an exercise of peer-learning via the online platform (discussion forum). You are expected to formulate your solution using this mode. You are to participate by making postings for ALL items listed below.

- at least **TWO (2)** difficulties or questions you have about the problem
- at least **TWO (2)** suggestions, clarifications, observation or feedback to other students' postings, to be accompanied with reasons
- at least **ONE (1)** important lesson for EACH aspect of this problem:
 - inheritance and object composition,
 - abstract classes and abstract methods
 - instance variables (or object attributes) and class variables,
 - instance (or object) methods and class methods and overriding methods.

Simply giving definition of these terms is insufficient. You must apply the terms to the problem described in the question.

Copy and paste at least THREE (3) postings that collectively cover ALL bulleted items. The postings you submit should have at least THREE (3) different posting dates on THREE (3) different weeks. Each posting should not cover more than 1 of the 3 bulleted items. If postings are made on the same week (Sun-Sat), only ONE (1) of them will be awarded mark.

(10 marks)

- (b) Develop the specification by describing the classes and their relationships. Note that you are **NOT** required to write Python code.
- Highlight the relationships between classes such as inheritance and object composition, and abstract superclasses, if any.
 - Suggest a few instance (or object) variables and class variables, instance (or object) methods, class methods, abstract methods and overriding methods, if applicable, for each class.
 - Justify your design by giving reason(s) for your design.

(10 marks)

For part (a), write your learning journey using the above guideline to document your learning experience. You are to copy and paste screenshots of your postings to the word document. Make sure your screenshots of your postings are readable. Copy also the links to each of your postings.

In addition, write your answers to part (b) to the word document. You may supplement your answer with a class diagram, an example of which is shown in Figure Q1.

Question 3

A delivery company provides both local and overseas deliveries which are either normal delivery or express delivery. Every sender must be registered. To have an item delivered, the sender calls up the company to request for the item to be picked up. The destinations of overseas deliveries are classified into 6 zones.

Study the classes shown in Figure Q3 for the delivery application.

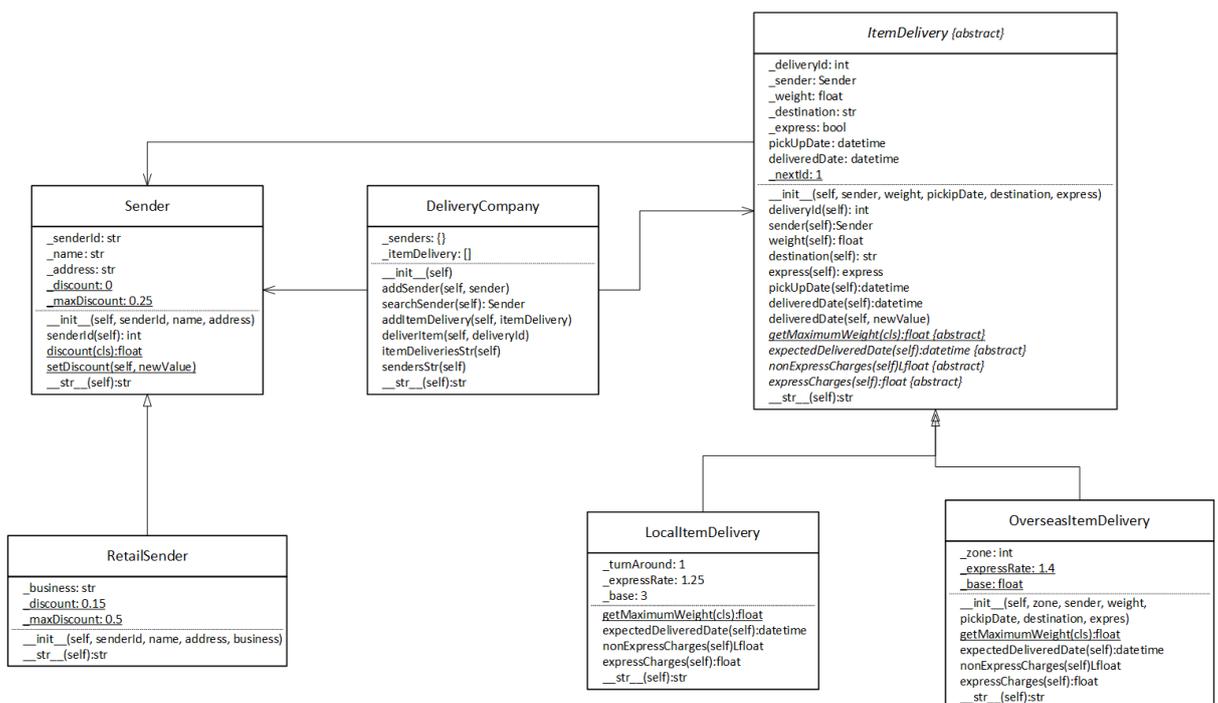


Figure Q3a

You are to write the following classes:

- (a) **DeliveryException** class, a subclass of **Exception**

When the application encounters a business rule violation, an exception from this class is raised.

(1 mark)

- (b) **Sender** class

- There are three object attributes: sender id, name and address. There are two class variables,
 - discount which currently is 0. Currently, there is no discount given to senders.
 - the maximum discount that any sender may have. Currently the value is 0.25

There is a constructor with three parameters to initialise the object attributes, and an accessor method for sender id.

There are accessor and mutator method for only the class variable, discount.

The mutator method for discount raises a **DeliveryException** when an attempt is made to set the discount to a negative value or to a value greater than the maximum discount that the delivery company is willing to give, with the message `Discount given cannot be negative` or `Discount given cannot be greater than xx` respectively, where `xx` is the maximum discount.

- A method `__str__` returns a string representation of a sender. An example is shown here:

```
Sender Id: S123 Name: Peter Tan Address: 12 Tampines
```

(4 marks)

- (c) **RetailSender** class which is a subclass of **Sender**

- Beside the same object attributes as those in **Sender**, that is, sender Id, name and address, there is an additional object attribute, the business of the retailer.

There are two class variables,

- discount which currently is 0.15, as retailers currently get 15% off the delivery charges
- the maximum discount that any retailer may get. Currently the value is 0.5 or 50%.

There is a constructor with four parameters to initialise all four object attributes.

- A method `__str__` returns a string representation of a retail sender. An example is shown here:

```
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
```

(2 marks)

- (d) **ItemDelivery** class is abstract superclass

- There are seven object attributes:
 - the delivery id,

- the sender requesting for the delivery,
- the weight of the item,
- the pickup date,
- the destination,
- whether the delivery is express (True) or normal(False), and
- the delivered date.

There is one class attribute in this class, `_nextId` which has the value 1. `_nextId` is used to auto generate running numbers for the delivery id. The delivery id for the first delivery is 1, the second is 2, and so on.

There is a constructor with five parameters to initialise five object attributes excluding the delivery id and delivered date. The delivered date is set to None. It is set only when the item reaches its destination. The constructor raises a **DeliveryException** when an attempt is made to set the weight to a negative value or a value greater than the maximum weight allowed for a delivery, with the message `Item weight xx cannot be negative or Item weight xxkg exceed maximum weight yykg` respectively, where `xx` is weight of the item and `yy` is the maximum allowed weight.

There are accessor method for all the object attributes, and a mutator method for delivered date.

- There are 4 abstract methods, of which one is an abstract class method.
 - `getMaximumWeight` returns the maximum weight allowed for an item that is to be delivered. This is an abstract class method that should be called by the constructor.
 - `expectedDeliveredDate` returns the expected date that the item will reach its destination. This is an abstract method.
 - `nonExpressCharges` returns the delivery charges for the item if delivered by normal delivery. This is an abstract method.
 - `expressCharges` returns the delivery charges for the item if delivered by express delivery. This is an abstract method.
- A `charges` method returns the charges for normal or express delivery if a discount does not apply, and it returns the discounted charges if a discount applies.
- A method `__str__` returns a string representation of the delivery. Two examples are shown below, one showing an item has been delivered and the other showing that the item is still in transit.

```

Delivery Id: 1
Express: Yes      weight: 4kg      Charges: $12.50
Pick Up: Thu, 05 Sep 2019      Expected delivery date: Thu, 05 Sep 2019
Collected: Thu, 05 Sep 2019      Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Delivery Id: 2
Express: No      weight: 4kg      Charges: $10.00
Pick Up: Thu, 05 Sep 2019      Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit      Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

```

(5 marks)

(e) **LocalItemDelivery** class is a subclass of **ItemDelivery**

- Besides having the same 7 object attributes as **ItemDelivery**, there is no additional object attribute for **LocalItemDelivery**. However, there are three class attributes.
 - express rate is 1.25, that is it costs 25% more to deliver in express mode.
 - the turnaround time is 1 day for normal delivery,
 - the base rate is \$3 which must be added to the charges.
- The class implements each of the 4 abstract methods.
 - `getMaximumWeight` returns 40. The weight limit is 40kg.
 - the `expectedDeliveredDate` for normal delivery is the pickup date added with the turnaround time. For expressed delivery, the turnaround time is halved.
 - `nonExpressCharges` returns the charges for an item with normal delivery, according to Table Q3a:

Weight of item	Charges
< 3 kg	\$3 + base rate
3 to less than 5kg	\$7 + base rate
5kg to less 10kg	\$12 + base rate
10kg or more	\$30 + base rate

Table Q3a

- `expressCharges` is computed by multiplying the express rate and the charges for normal delivery.
- A method `__str__` returns a string representation of a local delivery. Four examples are shown below:

```
Local Item Delivery Delivery Id: 1
Express: Yes    weight: 4kg    Charges: $12.50
Pick Up: Thu, 05 Sep 2019    Expected delivery date: Thu, 05 Sep 2019
Collected: Thu, 05 Sep 2019    Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Local Item Delivery Delivery Id: 2
Express: No    weight: 4kg    Charges: $10.00
Pick Up: Thu, 05 Sep 2019    Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit    Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Local Item Delivery Delivery Id: 5
Express: Yes    weight: 4kg    Charges: $10.62
Pick Up: Thu, 05 Sep 2019    Expected delivery date: Thu, 05 Sep 2019
Collected: Thu, 05 Sep 2019    Destination: 1 Delivery Road
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Local Item Delivery Delivery Id: 6
Express: No    weight: 4kg    Charges: $8.50
Pick Up: Thu, 05 Sep 2019    Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit    Destination: 1 Delivery Road
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
```

(5 marks)

(f) **OverseasItemDelivery** class is a subclass of **ItemDelivery**

- Besides the same 7 object attributes as **ItemDelivery**, there is one additional object attribute, zone. and two class attributes.
 - express rate is 1.4, that is it costs 40% more to deliver overseas in express mode.
 - the base rate is \$10

The constructor raises a **DeliveryException** when an attempt is made to set the zone to a negative value or to a value greater than 6 with the message zone must be between 1 and 6.

- The class implements each of the 4 abstract methods.
 - `getMaximumWeight` returns 30. The weight limit is 30kg.
 - `expectedDeliveredDate` for normal delivery is the pickup date added with the turnaround time which is 3 times the zone of an overseas destination. For express delivery, the turnaround time is 2 times the zone of an overseas destination. For example, the delivery to zone 1 has a turnaround time of 3 days for normal delivery and 2 days for express delivery.
 - `nonExpressCharges` which returns charges for an item with normal delivery, according to Table Q3b:

Weight of item	Charges
< 3 kg	\$30 + base rate x zone/10
3 to less than 5kg	\$50 + base rate x zone/10
5kg to less 10kg	\$90 + base rate x zone/10
10kg or more	\$150 + base rate x zone/10

Table Q3b

- `expressCharges` returns charges for the item with express delivery. `expressCharges` returns the value obtained by multiplying the sum of the express rate and zone/10, with the charges for normal delivery.
- A method `__str__` returns a string representation of an overseas delivery.

Four examples are shown below:

```
Overseas Item Delivery Zone: 1 Delivery Id: 7
Express: Yes   weight: 4kg   Charges: $65.02
Pick Up: Thu, 05 Sep 2019   Expected delivery date: Sat, 07 Sep 2019
Collected: Not collected, in transit   Destination: 1 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Overseas Item Delivery Zone: 1 Delivery Id: 8
Express: No   weight: 4kg   Charges: $43.35
Pick Up: Thu, 05 Sep 2019   Expected delivery date: Sun, 08 Sep 2019
Collected: Not collected, in transit   Destination: 1 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Overseas Item Delivery Zone: 2 Delivery Id: 10
Express: No   weight: 4kg   Charges: $44.20
Pick Up: Thu, 05 Sep 2019   Expected delivery date: Wed, 11 Sep 2019
Collected: Not collected, in transit   Destination: 2 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Overseas Item Delivery Zone: 2 Delivery Id: 9
Express: Yes   weight: 4kg   Charges: $70.72
Pick Up: Thu, 05 Sep 2019   Expected delivery date: Mon, 09 Sep 2019
Collected: Not collected, in transit   Destination: 2 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
```

(5 marks)

(g) **DeliveryCompany** class

- There are two object attributes: a collection of senders (dict) and a collection of item deliveries (list).
There is a constructor with no parameter, and both object attributes are set to empty collections.
- A `searchSender` method returns a sender in the collection that has the same sender id as the parameter. The method returns `None` if the search is unsuccessful.
- There are two `add` methods, one for each of the object attributes
 - An `addSender` method that has one parameter, `sender`. It adds a sender to the collection only if the collection does not contain a sender with the same sender id. The method raises a **DeliveryException** with the message `Sender xx is already recorded` where `xx` is the sender id.
 - An `addDeliveryItem` method that adds a delivery to the collection
- A `deliverItem` method that has one parameter, a delivery id. The method raises a **DeliveryException** if the delivery id does not match any of the deliveries in the collection. The exception is raised with exception message `Invalid delivery Id xx` where `xx` is the delivery id. Otherwise, the method sets the delivered date of the located delivery to today's date, and returns `True`.
- There are two `str` methods for object attributes.
 - A `itemDeliveriesStr` method returns a string with each item delivery detail on separate lines.
 - A `sendersStr` method returns a string with each sender detail on separate lines.
- The `str` method returns the object attributes as a string in the format shown here:

Senders:

Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Deliveries:

Local Item Delivery Delivery Id: 1
Express: Yes weight: 4kg Charges: \$12.50
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Thu, 05 Sep 2019
Collected: Thu, 05 Sep 2019 Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Local Item Delivery Delivery Id: 2
Express: No weight: 4kg Charges: \$10.00
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Local Item Delivery Delivery Id: 5
Express: Yes weight: 4kg Charges: \$10.62
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Thu, 05 Sep 2019
Collected: Thu, 05 Sep 2019 Destination: 1 Delivery Road
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Local Item Delivery Delivery Id: 6
Express: No weight: 4kg Charges: \$8.50
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit Destination: 1 Delivery Road
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Overseas Item Delivery Zone: 1 Delivery Id: 3
Express: Yes weight: 4kg Charges: \$76.50

Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sat, 07 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 4
 Express: No weight: 4kg Charges: \$51.00
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sun, 08 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 7
 Express: Yes weight: 4kg Charges: \$65.02
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sat, 07 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 8
 Express: No weight: 4kg Charges: \$43.35
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sun, 08 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 2 Delivery Id: 10
 Express: No weight: 4kg Charges: \$44.20
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Wed, 11 Sep 2019
 Collected: Not collected, in transit Destination: 2 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 2 Delivery Id: 9
 Express: Yes weight: 4kg Charges: \$70.72
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Mon, 09 Sep 2019
 Collected: Not collected, in transit Destination: 2 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

(5 marks)

(h) The application performs the following tasks:

- Create a **DeliveryCompany** object and populate it with the 2 senders, 4 local deliveries and 4 overseas deliveries. You may change the pickup dates to any date that helps demonstrate that your program is correct.
- The application must handle all exceptions including input error. Input error should be handled by allowing user to re-enter. Unless specified, simply handle **DeliveryException** raised by printing the error message. For example,

```
Menu
1. List Senders
2. List Deliveries
3. Add Sender
4. Request Delivery Pickup
5. Deliver Item
0. Exit
Enter option: a
Please enter a whole number
Menu
1. List Senders
2. List Deliveries
3. Add Sender
4. Request Delivery Pickup
5. Deliver Item
0. Exit
Enter option:
Please enter a whole number
```

- Provide a menu to allow a user to repeatedly select an option from the menu to perform an operation on the **DeliveryCompany** object.

```
Menu
```

1. List Senders
 2. List Deliveries
 3. Add Sender
 4. Request Delivery Pickup
 5. Deliver Item
 0. Exit
- Perform the following actions for each of the options:

Option 1: Display the collection of senders

Example:

Enter option: 1
 Sender List:
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Option 2: Display the collection of deliveries

Example:

Enter option: 2
 Delivery List:
 Local Item Delivery Delivery Id: 1
 Express: Yes weight: 4kg Charges: \$12.50
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Thu, 05 Sep 2019
 Collected: Not collected, in transit Destination: 1 Delivery Road
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines
 Local Item Delivery Delivery Id: 2
 Express: No weight: 4kg Charges: \$10.00
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Fri, 06 Sep 2019
 Collected: Not collected, in transit Destination: 1 Delivery Road
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines
 Local Item Delivery Delivery Id: 5
 Express: Yes weight: 4kg Charges: \$10.62
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Thu, 05 Sep 2019
 Collected: Not collected, in transit Destination: 1 Delivery Road
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Local Item Delivery Delivery Id: 6
 Express: No weight: 4kg Charges: \$8.50
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Fri, 06 Sep 2019
 Collected: Not collected, in transit Destination: 1 Delivery Road
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 3
 Express: Yes weight: 4kg Charges: \$76.50
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sat, 07 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 4
 Express: No weight: 4kg Charges: \$51.00
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sun, 08 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 7
 Express: Yes weight: 4kg Charges: \$65.02
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sat, 07 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 1 Delivery Id: 8
 Express: No weight: 4kg Charges: \$43.35
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sun, 08 Sep 2019
 Collected: Not collected, in transit Destination: 1 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 2 Delivery Id: 10
 Express: No weight: 4kg Charges: \$44.20
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Wed, 11 Sep 2019
 Collected: Not collected, in transit Destination: 2 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
 Overseas Item Delivery Zone: 2 Delivery Id: 9
 Express: Yes weight: 4kg Charges: \$70.72

Pick Up: Thu, 05 Sep 2019 Expected delivery date: Mon, 09 Sep 2019
 Collected: Not collected, in transit Destination: 2 Some Land
 Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines

Option 3:

Prompt for the sender's detail and the type of sender, and add the sender to the collection. Exception raised due to duplicate sender id should be handled by allowing the user 2 more attempts to re-enter the sender id or an empty string to abandon option 3.

Example 1:

```
Enter option: 3
Enter sender identification: s123
Enter sender name: test
Enter address: 1 Road
Enter type of sender - 1 for individual and 2 for retail: 1
Sender S123 is already recorded
You have two attempts to correct sender identification
Attempt 1 - Try another sender identification or <ENTER> to abandon: T111
Sender T111 is already recorded
Attempt 2 - Try another sender identification or <ENTER> to abandon: S123
Sender S123 is already recorded
Cannot add sender
```

Example 2:

```
Enter option: 3
Enter sender identification: T111
Enter sender name: Lee and Partner
Enter address: 51 Clememti Road
Enter type of sender - 1 for individual and 2 for retail: 2
Enter type of goods being retailed: Toy
Sender T111 is already recorded
You have two attempts to correct sender identification
Attempt 1 - Try another sender identification or <ENTER> to abandon: T222
Retail Business Toy Sender Id: T222 Name: Lee and Partner Address: 51 Clememti Road added!
```

Example 3:

```
Enter option: 3
Enter sender identification: S234
Enter sender name: John
Enter address: 36 Joo Chiat
Enter type of sender - 1 for individual and 2 for retail: 1
Sender Id: S234 Name: John Address: 36 Joo Chiat added
```

Example 4:

```
Enter option: 3
Enter sender identification: s123
Enter sender name: tom
Enter address: 13 Haig Road
Enter type of sender - 1 for individual and 2 for retail: 1
Sender S123 is already recorded
You have two attempts to correct sender identification
Attempt 1 - Try another sender identification or <ENTER> to abandon:
Sender cannot added due to duplicate identification
```

Option 4:

Prompt for the sender id, and handle exception by allowing the user 3 attempts to enter the sender id that belongs to a registered sender. If a sender id is correct, prompt for item detail, and add a delivery to the collection.

Example 1:

```
Enter option: 4
```

Please enter correct sender id within 3 attempts
 1. Enter sender id <ENTER> to end: 1
 Invalid sender Id
 2. Enter sender id <ENTER> to end: 2
 Invalid sender Id
 3. Enter sender id <ENTER> to end: 3
 Invalid sender Id
 Please register the sender first

Example 2:

Enter option: 4
 Please enter correct sender id within 3 attempts
 1. Enter sender id <ENTER> to end: a
 Invalid sender Id
 2. Enter sender id <ENTER> to end: s123
 Enter weight of item to deliver: 3
 Enter pick up date in d/m/YYYY format: 5/9/2019
 Enter destination address: somewhere
 By express? y for yes: y
 Enter type of delivery - 1 for local and 2 for overseas: 2
 Enter zone of destination: 3
 Overseas Item Delivery Zone: 3 Delivery Id: 11
 Express: Yes weight: 3.0kg Charges: \$90.10
 Pick Up: Thu, 05 Sep 2019 Expected delivery date: Wed, 11 Sep 2019
 Collected: Not collected, in transit Destination: somewhere
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines added

Example 3:

Enter option: 4
 Please enter correct sender id within 3 attempts
 1. Enter sender id <ENTER> to end: s123
 Enter weight of item to deliver: 0
 Please enter a positive number for weight
 Enter weight of item to deliver:
 Please enter a number
 Enter weight of item to deliver: a
 Please enter a number
 Enter weight of item to deliver: 0.1
 Enter pick up date in d/m/YYYY format: 1
 Please enter pick up date in d/m/YYYY format
 Enter pick up date in d/m/YYYY format: 12 12 2019
 Please enter pick up date in d/m/YYYY format
 Enter pick up date in d/m/YYYY format: 12/12/2019
 Enter destination address: testing
 By express? y for yes: q
 Enter type of delivery - 1 for local and 2 for overseas: 0
 Please enter either 1 for local or 2 for overseas
 Enter type of delivery - 1 for local and 2 for overseas: 3
 Please enter either 1 for local or 2 for overseas
 Enter type of delivery - 1 for local and 2 for overseas: -1
 Please enter either 1 for local or 2 for overseas
 Enter type of delivery - 1 for local and 2 for overseas: 1
 Local Item Delivery Delivery Id: 11
 Express: No weight: 0.1kg Charges: \$6.00
 Pick Up: Thu, 12 Dec 2019 Expected delivery date: Fri, 13 Dec 2019
 Collected: Not collected, in transit Destination: testing
 Sender Id: S123 Name: Peter Tan Address: 12 Tampines added

Option 5:

Prompt for a delivery id, and update the delivered date to today's date. List the delivery company object if the update is successful to demonstrate that your update is correctly performed.

Example 1:

Enter option: 5
 Enter delivery id: 0
 Invalid delivery Id 0

Example 2:

```

Enter option: 5
Enter delivery id: 1
Senders:
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

Deliveries:
Local Item Delivery Delivery Id: 1
Express: Yes weight: 4kg Charges: $12.50
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Thu, 05 Sep 2019
Collected: Thu, 05 Sep 2019 Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines
Local Item Delivery Delivery Id: 2
Express: No weight: 4kg Charges: $10.00
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit Destination: 1 Delivery Road
Sender Id: S123 Name: Peter Tan Address: 12 Tampines
Local Item Delivery Delivery Id: 5
Express: Yes weight: 4kg Charges: $10.62
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Thu, 05 Sep 2019
Collected: Not collected, in transit Destination: 1 Delivery Road
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Local Item Delivery Delivery Id: 6
Express: No weight: 4kg Charges: $8.50
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Fri, 06 Sep 2019
Collected: Not collected, in transit Destination: 1 Delivery Road
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Overseas Item Delivery Zone: 1 Delivery Id: 3
Express: Yes weight: 4kg Charges: $76.50
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sat, 07 Sep 2019
Collected: Not collected, in transit Destination: 1 Some Land
Sender Id: S123 Name: Peter Tan Address: 12 Tampines
Overseas Item Delivery Zone: 1 Delivery Id: 4
Express: No weight: 4kg Charges: $51.00
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sun, 08 Sep 2019
Collected: Not collected, in transit Destination: 1 Some Land
Sender Id: S123 Name: Peter Tan Address: 12 Tampines
Overseas Item Delivery Zone: 1 Delivery Id: 7
Express: Yes weight: 4kg Charges: $65.02
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sat, 07 Sep 2019
Collected: Not collected, in transit Destination: 1 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Overseas Item Delivery Zone: 1 Delivery Id: 8
Express: No weight: 4kg Charges: $43.35
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Sun, 08 Sep 2019
Collected: Not collected, in transit Destination: 1 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Overseas Item Delivery Zone: 2 Delivery Id: 10
Express: No weight: 4kg Charges: $44.20
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Wed, 11 Sep 2019
Collected: Not collected, in transit Destination: 2 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Overseas Item Delivery Zone: 2 Delivery Id: 9
Express: Yes weight: 4kg Charges: $70.72
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Mon, 09 Sep 2019
Collected: Not collected, in transit Destination: 2 Some Land
Retail Business Garment Sender Id: T111 Name: Tan and Family Address: 15 Tampines
Overseas Item Delivery Zone: 3 Delivery Id: 11
Express: Yes weight: 3.0kg Charges: $90.10
Pick Up: Thu, 05 Sep 2019 Expected delivery date: Wed, 11 Sep 2019
Collected: Not collected, in transit Destination: somewhere
Sender Id: S123 Name: Peter Tan Address: 12 Tampines

```

(13 mark)

Copy and paste the python code as a single program to the word document. Copy and paste ONE (1) screenshot - one screenshot showing the output from your program for options 1 to 5 for the various scenarios.

Question 4

Write a GUI delivery calculator that compute charges based on volumetric weight or actual weight of item, whichever is higher.

The volumetric weight in kg is $\frac{\text{length} \times \text{width} \times \text{height}}{6000}$ if dimensions are measured in cm

The volumetric weight in kg is $\frac{\text{length} \times \text{width} \times \text{height}}{366}$ if dimensions are measured in inches

Use Table Q3a in Question 3 to compute the delivery charges based on the higher of the weights, that is, the volumetric weight and actual weight of item.

- (a) Implement the GUI as shown in Figure Q4a. The title of the GUI MUST include your name.

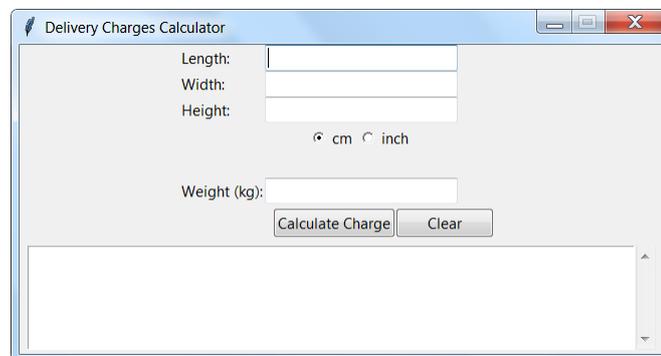


Figure Q4a The initial GUI

(7 marks)

- (b) You do not need to handle exceptions for this question. Implement event handling for each of the 2 buttons.
- Calculate Charge

The application calculates the volumetric weight of the item based on the input in the text fields, and compares it with the actual weight of the item. Taking the higher of the two, the application then displays the delivery charges. Figure Q4b and Figure Q4c show the effect of user interactions on the GUI. Note that the output is **appended** to the scroll pane to allow the user to review the various item charges.

Figure Q4b Before Calculate Charge button is clicked

Figure Q4c After Calculate Charge button is clicked

- Clear

The application clears both the data in the text fields as well as the output in the scroll pane. The GUI reverts to the one shown in Figure Q4a.

(8 marks)

Copy and paste the python code for Q4a and Q4b as a single application to the word document. Copy and paste ONE (1) screenshot of your GUI.

---- END OF ASSIGNMENT ----