Isser Troy M. Gagan

11843624

**MACHINE PROJECT (Phase 2)**

**TEST CASES**

Class: Machine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| trackParcel | 1 | Determines the current date, tracking number, date of transaction, and ETA of a parcel transaction when the tracking number input is incorrect. | > Tracking number: BUX0809LEZ0101 | “Invalid tracking number, transaction not found.” | “Invalid tracking number, transaction not found.” | P |
|  | 2 | Determines the current date, tracking number, date of transaction, and ETA of a parcel transaction when the tracking number input is correct (assuming the transaction exists). | > Tracking number: BOX0809LUZ01001 | > Current Date: August 09, 2019  > Tracking Number: BOX0809LUZ01001  > Date of transaction: August 09, 2019  > ETA: August 11, 2019 | > Current Date: August 09, 2019  > Tracking Number: BOX0809LUZ01001  > Date of transaction: August 09, 2019  > ETA: August 11, 2019 | P |
|  | 3 | Determines the current date, tracking number, date of transaction, and ETA of a parcel transaction when the tracking number input is correct and has a mix of lowercase and uppercase letters. | > Tracking number: boX0809LuZ01001 | > Current Date: August 09, 2019  > Tracking Number: BOX0809LUZ01001  > Date of transaction: August 09, 2019  > ETA: August 11, 2019 | > Current Date: August 09, 2019  > Tracking Number: BOX0809LUZ01001  > Date of transaction: August 09, 2019  > ETA: August 11, 2019 | P |
| getStatusOfParcel | 1 | Determines the status of the parcel transaction when the tracking number input is incorrect. | > Tracking number: FTL9988MML0208 | \*returns an empty string | \*returns an empty string | P |
|  | 2 | Determines the status of the parcel transaction when the tracking number input is correct (assuming the transaction exists) and the current date is August 9, 2019. | > Tracking number: FLT0809VIS01001 | > Status: Preparing | > Status: Preparing | P |
|  | 3 | Determines the status of the parcel transaction when the tracking number input is correct (assuming the transaction exists) and the current date is August 10, 2019. | > Tracking number: FLT0809VIS01001 | > Status: Shipping | > Status: Shipping | P |
|  | 4 | Determines the status of the parcel transaction when the tracking number input is correct (assuming the transaction exists) and the current date is August 13, 2019. | > Tracking number: FLT0809VIS01001 | > Status: Delivered | > Status: Delivered | P |
|  | 5 | Determines the status of the parcel transaction when the tracking number input is correct (assuming the transaction exists), the current date is August 14, 2019, and the input has a mix of lowercase and uppercase letters. | > Tracking number: fLT0809vIs01001 | > Status: Delivered | > Status: Delivered | P |
| getParcelOptions | 1 | Determines which of the (6) parcel sizes/types may be used for the items if the dimensions of the items exceed the parcel sizes/types. | Item # 1 (Regular-shaped product)  > length: 999  > width: 12  > height: 8  > weight (grams): 1000  Item # 2 (Document)  > length: 11  > width: 8.5  > number of pages: 26 | *Parcel Options*  > small pouch: unavailable  > large pouch: unavailable  > box 1: unavailable  > box 2: unavailable  > box 3: unavailable  > box 4: unavailable | *Parcel Options*  > small pouch: unavailable  > large pouch: unavailable  > box 1: unavailable  > box 2: unavailable  > box 3: unavailable  > box 4: unavailable | P |
|  | 2 | Determines which of the (6) parcel sizes/types may be used for the input item dimensions. | Item # 1 (Irregular-shaped product)  > length: 9  > width: 12  > height: 8  > weight (grams): 1000  Item # 2 (Document)  > length: 11  > width: 8.5  > number of pages: 25 | *Parcel Options*  > small pouch: unavailable  > large pouch: unavailable  > box 1: unavailable  > box 2: unavailable  > box 3: available  > box 4: available | *Parcel Options*  > small pouch: unavailable  > large pouch: unavailable  > box 1: unavailable  > box 2: unavailable  > box 3: available  > box 4: available | P |
|  | 3 | Determines which of the (6) parcel sizes/types may be used for the input item dimensions. | Item # 1 (Document)  > length: 9  > width: 14  > number of pages: 25 | *Parcel Options*  > small pouch: available  > large pouch: available  > box 1: unavailable  > box 2: available  > box 3: available  > box 4: available | *Parcel Options*  > small pouch: available  > large pouch: available  > box 1: unavailable  > box 2: available  > box 3: available  > box 4: available | P |
| isItemFit | 1 | Determines if an item fits a small pouch (9 x 14 x 1 inches). | Regular-shaped product  > length: 8  > width: 10  > height: 1  > weight (grams): 1000 | true | true | P |
|  | 2 | Determines if an item fits a large pouch (12 x 18 x 3 inches). | Document  > length: 14  > width: 20  > number of pages: 25 | false | false | P |
|  | 3 | Determines if an item fits a large pouch (12 x 18 x 3 inches) despite exceeding the maximum weight. | Irregular-shaped product  > length: 8  > width: 10  > height: 13  > weight (grams): 4000 | true | true | P |
| isAnyParcelAvailable | 1 | Determines if any parcel size/type is available depending on results of the getParcelOptions() method. | *Parcel Options*  > small pouch: unavailable  > large pouch: unavailable  > box 1: unavailable  > box 2: unavailable  > box 3: unavailable  > box 4: unavailable | false | false | P |
|  | 2 | Determines if any parcel size/type is available depending on results of the getParcelOptions() method. | *Parcel Options*  > small pouch: unavailable  > large pouch: unavailable  > box 1: unavailable  > box 2: available  > box 3: available  > box 4: available | true | true | P |
|  | 3 | Determines if any parcel size/type is available depending on results of the getParcelOptions() method | *Parcel Options*  > small pouch: available  > large pouch: available  > box 1: available  > box 2: available  > box 3: available  > box 4: available | true | true | P |
| createAccount | 1 | Determines the status of account creation when the username is not yet taken, the password and confirm password is the same, and the admin key is correct (assuming the correct admin key is: admin1234). | > username: issergagan  > password: ssrggn0813  > confirm password: ssrggn0813  > admin key: admin1234 | “Account has been successfully created.” | “Account has been successfully created.” | P |
|  | 2 | Determines the status of account creation when the username is taken, the password and confirm password is the same, and the admin key is correct (assuming the correct admin key is: admin1234). | > username: annecurtis  > password: ssrggn0813  > confirm password: ssrggn0813  > admin key: admin1234 | "Username already exists." | "Username already exists." | P |
|  | 3 | Determines the status of account creation when the username is not yet taken, the password and confirm password is not the same, and the admin key is correct (assuming the correct admin key is: admin1234). | > username: issergagan  > password: ssrggn0813  > confirm password: cocacola  > admin key: admin1234 | "Input password does not match confirm password." | "Input password does not match confirm password." | P |
|  | 4 | Determines the status of account creation when the username is not yet taken, the password and confirm password is the same, and the admin key is incorrect (assuming the correct admin key is: admin1234). | > username: issergagan  > password: ssrggn0813  > confirm password: ssrggn0813  > admin key: admin123456789 | "Wrong admin key, please ask admin for assistance." | "Wrong admin key, please ask admin for assistance." | P |
|  | 5 | Determines the status of account creation when there is no username and password input, and the admin key is correct (assuming the correct admin key is: admin1234). | > username:  > password:  > confirm password: ssrggn0813  > admin key: admin1234 | "No entry, please enter username and/or password." | "No entry, please enter username and/or password." | P |
| changePassword | 1 | Determines the status of the change password operation when the account exists, and the new password and confirm new password input is the same. | > username: issergagan  > old password: ssrggn0813  > new password: 12345678  > confirm new password: 12345678 | "Password has been changed successfully." | "Password has been changed successfully." | P |
|  | 2 | Determines the status of the change password operation when the account exists, and the new password and confirm new password input is the not same. | > username: issergagan  > old password: ssrggn0813  > new password: 12345678  > confirm new password: 12340000 | “Error: New password does not match confirm new password.” | “Error: New password does not match confirm new password.” | P |
|  | 3 | Determines the status of the change password operation when the account does not exists, and the new password and confirm new password input is the same. | > username: ajejeMon  > old password: ssrggn0813  > new password: 12345678  > confirm new password: 12345678 | “Warning: Incorrect username or password." | “Warning: Incorrect username or password." | P |
|  | 4 | Determines the status of the change password operation when the account exists, and the old password and the new password is the same. | > username: issergagan  > old password: ssrggn0813  > new password: ssrggn0813  > confirm new password: ssrggn0813 | "Error: Entered new password is the same as old password." | "Error: Entered new password is the same as old password." | P |
|  | 5 | Determines the status of the change password operation when the account exists, and the new password is empty. | > username: issergagan  > old password: ssrggn0813  > new password:  > confirm new password: | "Error: No new password input." | "Error: No new password input." | P |
| isAuthorizeLogin | 1 | Determines whether the account of the inputted username and password exists or is authorized, when the username and password belongs to an account. | > username: issergagan  > password: 12345678 | true | true | P |
|  | 2 | Determines whether the account of the inputted username and password exists or is authorized, when the username and password does not belongs to an account. | > username: issergaganSKSKSK  > password: 12345678 | false | false | P |
|  | 3 | Determines whether the account of the inputted username and password exists or is authorized, when the username and password belongs to an account but the input has incorrect case for the letters (uppercase/lowercase). | > username: iSseRgaGan  > password: 12345678 | false | false | P |

Class: TrackingInfo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| TrackingInfo (class constructor also creates the tracking number) | 1 | Determines the tracking number based on the specifications of the parcel (type, destination, etc…) and the current date. *Note: parcel index 0 is small pouch, 1 is large pouch, and indices 2 to 5 are box parcels.* | > parcel index: 0 (small pouch)  > transaction date: August 9, 2019  > destination: Metro Manila  > number of items: 1  > parcel number (sequence): 1 | > Tracking number: FLT0809MML01001 | > Tracking number: FLT0809MML01001 | P |
|  | 2 | Determines the tracking number based on the specifications of the parcel (type, destination, etc…) and the current date. | > parcel index: 3 (box)  > transaction date: August 9, 2019  > destination: Provincial Luzon  > number of items: 2  > parcel number (sequence): 2 | > Tracking number: BOX0809LUZ02002 | > Tracking number: BOX0809LUZ02002 | P |
|  | 3 | Determines the tracking number based on the specifications of the parcel (type, destination, etc…) and the current date. | > parcel index: 1 (large pouch)  > transaction date: August 10, 2019  > destination: Visayas  > number of items: 1  > parcel number (sequence): 1 | > Tracking number: FLT0810VIS01001 | > Tracking number: FLT0810VIS01001 | P |
| updateStatus | 1 | Determines the current status of the parcel delivery when the current date and date of delivery is the same. | Current Date: August 10, 2019  > date of delivery: August 10, 2019 | > Status: Delivered | > Status: Delivered | P |
|  | 2 | Determines the current status of the parcel delivery depending on the current date and delivery date of the parcel. | Current Date: August 10, 2019  > date of transaction: August 10, 2019 | > Status: Preparing | > Status: Preparing | P |
|  | 3 | Determines the current status of the parcel delivery depending on the current date and delivery date of the parcel. | Current Date: August 10, 2019  > date of transaction: August 12, 2019  > date of delivery: August 15, 2019 | > Status: Shipping | > Status: Shipping | P |

Class: Transaction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| computeFees | 1 | Determines base rate fee, delivery fee, and insurance fee for the input parcel details. | > parcel: Flat (small pouch)  > destination: Metro Manila  > insure: Yes  *Item: Document*  > length: 5  > width: 4  > number of pages: 24 | > Base rate fee: Php 30.00  > Delivery fee: Php 50.00  > Insurance fee: Php 5.00 | > Base rate fee: Php 30.00  > Delivery fee: Php 50.00  > Insurance fee: Php 5.00 | P |
|  | 2 | Determines base rate fee, delivery fee, and insurance fee for the input parcel details. | > parcel: Box  > destination: Visayas  > insure: Yes  *Item 1: Regular product*  > length: 5  > width: 4  > height: 2  > weight (grams): 1000  *Item 2: Irregular product*  > length: 8  > width: 12  > height: 5  > weight (grams): 1000 | > Base rate fee: Php 80.00  > Delivery fee: Php 3000.00  > Insurance fee: Php 10.00 | > Base rate fee: Php 80.00  > Delivery fee: Php 3000.00  > Insurance fee: Php 10.00 | P |
|  | 3 | Determines base rate fee, delivery fee, and insurance fee for the input parcel details. | > parcel: Flat (large pouch)  > destination: Provincial Luzon  > insure: No  *Item: Document*  > length: 8.5  > width: 11  > number of pages: 51 | > Base rate fee: Php 50.0  > Delivery fee: Php 100.00  > Insurance fee: Php 0.00 | > Base rate fee: Php 50.0  > Delivery fee: Php 100.00  > Insurance fee: Php 0.00 | P |
| getTotalFee | 1 | Determines the total fee from the sum of the base rate fee, delivery fee, and insurance fee. | > Base rate fee: Php 30.00  > Delivery fee: Php 50.00  > Insurance fee: Php 5.00 | > Total fee: Php 85.00 | > Total fee: Php 85.00 | P |
|  | 2 | Determines the total fee from the sum of the base rate fee, delivery fee, and insurance fee. | > Base rate fee: Php 80.00  > Delivery fee: Php 3000.00  > Insurance fee: Php 10.00 | > Total fee: Php 3090.00 | > Total fee: Php 3090.00 | P |
|  | 3 | Determines the total fee from the sum of the base rate fee, delivery fee, and insurance fee. | > Base rate fee: Php 50.0  > Delivery fee: Php 100.00  > Insurance fee: Php 0.00 | > Total fee: Php 150.00 | > Total fee: Php 150.00 | P |

Class: Item

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| Item (constructor also sets the length, width, and actual weight of the item) | 1 | Determines if the item is created when the input length, width, and actual weight are all 0. | > length: 0  > width: 0  > actual weight: 0 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 2 | Determines if the item is created when the input length, width, and actual weight are all negative values. | > length: -1  > width: -7  > actual weight: -300 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 3 | Determines if the item is created when one of the inputs is 0. | > length: 3  > width: 5  > actual weight: 0 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 4 | Determines if the item is created when one of the inputs is a negative value. | > length: -8  > width: 5  > actual weight: 250 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 5 | Determines if the item is created when the length, width, and actual weight are non-zero and are positive values. | > length: 9  > width: 6  > actual weight: 3000 | *Item is created and no IllegalArgumentException is thrown.* | *Item is created and no IllegalArgumentException is thrown.* | P |

Class: Product

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| Product (constructor also sets the length, width, height, and actual weight of the product) | 1 | Determines if the product is created when the input length, width, height, and actual weight are all 0. | > length: 0  > width: 0  > height: 0  > actual weight: 0 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 2 | Determines if the product is created when the input length, width, height, and actual weight are all negative values. | > length: -6  > width: -2  > height: -5  > actual weight: -600 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 3 | Determines if the product is created when height is 0. | > length: 4  > width: 2  > height: 0  > actual weight: 2000 | IllegalArgumentException: "Product measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Product measurement/s cannot be a negative value or zero." | P |
|  | 4 | Determines if the product is created when height is a negative value. | > length: 6  > width: 5  > height: -4  > actual weight: 250 | IllegalArgumentException: "Product measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Product measurement/s cannot be a negative value or zero." | P |
|  | 5 | Determines if the product is created when the length, width, height, and actual weight are non-zero and are positive values. | > length: 4  > width: 9  > height: 2  > actual weight: 3000 | *Product is created and no IllegalArgumentException is thrown.* | *Product is created and no IllegalArgumentException is thrown.* | P |
| getSmallestMeasure | 1 | Determines the smallest measure of the product when the dimensions have different measures. | > length: 3  > width: 1  > height: 2 | > Smallest measure: 1 | > Smallest measure: 1 | P |
|  | 2 | Determines the smallest measure of the product when two of the dimensions have the same measure. | > length: 2  > width: 8  > height: 2 | > Smallest measure: 2 | > Smallest measure: 2 | P |
|  | 3 | Determines the smallest measure of the product when all of the dimensions have the same measure | > length: 5  > width: 5  > height: 5 | > Smallest measure: 5 | > Smallest measure: 5 | P |

Class: Document

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| Document (constructor also sets the length and width) | 1 | Determines if the document is created when the input length, width, height, and actual weight are all 0. | > length: 0  > width: 0  > number of pages: 0 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 2 | Determines if the document is created when the input length, width, height, and actual weight are all negative values. | > length: -5  > width: -3  > number of pages: -3 | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | IllegalArgumentException: "Item measurement/s cannot be a negative value or zero." | P |
|  | 3 | Determines if the document is created when number of pages is 0. | > length: 8  > width: 3  > number of pages: 0 | IllegalArgumentException: “Document measurement/s cannot be a negative value or zero." | IllegalArgumentException: “Document measurement/s cannot be a negative value or zero." | P |
|  | 4 | Determines if the document is created when number of pages is a negative value. | > length: 6  > width: 5  > number of pages: -52 | IllegalArgumentException: “Document measurement/s cannot be a negative value or zero." | IllegalArgumentException: “Document measurement/s cannot be a negative value or zero." | P |
|  | 5 | Determines if the document is created when the length, width, height, and actual weight are non-zero and are positive values. | > length: 4  > width: 9  > number of pages: 27 | *Document is created and no IllegalArgumentException is thrown.* | *Document is created and no IllegalArgumentException is thrown.* | P |
| getSmallestMeasure | 1 | Determines the smallest measure of the document when the dimensions have different measures. | > length: 4  > width: 1  > thickness: 7 | > Smallest measure: 1 | > Smallest measure: 1 | P |
|  | 2 | Determines the smallest measure of the document when two of the dimensions have the same measure. | > length: 2  > width: 8  > thickness: 4 | > Smallest measure: 2 | > Smallest measure: 2 | P |
|  | 3 | Determines the smallest measure of the document when all of the dimensions have the same measure | > length: 7  > width: 7  > thickness: 7 | > Smallest measure: 7 | > Smallest measure: 7 | P |
| Document (constructor also calculates the thickness, and actual weight of the product) | 1 | Determines the thickness and actual weight of the document when the number of pages is less than 25. | > number of pages: 20 | > thickness: 1  > actual weight: 1000 grams (1 kilo) | > thickness: 1  > actual weight: 1000 grams (1 kilo) | P |
|  | 2 | Determines the thickness and actual weight of the document when the number of pages is 25. | > number of pages: 25 | > thickness: 1  > actual weight: 1000 grams (1 kilo) | > thickness: 1  > actual weight: 1000 grams (1 kilo) | P |
|  | 3 | Determines the thickness and actual weight of the document when the number of pages is more than 25. | > number of pages: 28 | > thickness: 2  > actual weight: 1000 grams (1 kilo) | > thickness: 2  > actual weight: 1000 grams (1 kilo) | P |

Class: IrregularShapedProduct

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | # | Test Description | Sample Input Data | Expected Output | Actual Output | Pass/Fail |
| IrregularShapedProduct(constructor also calculates the volumetric weight) | 1 | Determines the volumetric weight based on the length, width, and height input. | > length: 1  > width: 2  > height: 3 | > volumetric weight (grams): 19.67 | > volumetric weight (grams): 19.67 | P |
|  | 2 | Determines the volumetric weight based on the length, width, and height input. | > length: 8  > width: 3  > height: 5 | > volumetric weight (grams): 393.44 | > volumetric weight (grams): 19.67 | P |
|  | 3 | Determines the volumetric weight based on the length, width, and height input. | > length: 5  > width: 6  > height: 2 | > volumetric weight (grams): 196.72 | > volumetric weight (grams): 19.67 | P |