Start

MainMenu.menu(args);

Declare variable for menu.

Display menu.

DO WHILE menu is not equal to 1 and menu not equal to 2 and menu not equal to 3 and menu not equal to 4 and menu not equal to 5

Print “Enter the menu you want:”.

Input menu.

END DO

IF menu is equal to 1 THEN

Call method Poslaju.main(args).

ELSE IF menu is equal to 2 THEN

Call method Samedaydelivery.main(args).

ELSE IF menu is equal to 3 THEN

Call method PrepaidBEnvelope.main(args).

ELSE IF menu is equal to 4 THEN

Call method PosEkspres.main(args).

ELSE THEN

Exit.main(args).

END IF

Return main().

Poslaju.main(args)

Output

Display the menu for Next-Day-Delivery

Display “Do you want to place order? 1 is YES,2 is NO”

Input

While user enter 1

Output

Display

1.Enter zone

2.Enter the weight of goods

3.Display the type of goods

4.Enter the quantity

Input

1.Zone

2.The weight of goods

3.Document/parcel

4.Quantity

If the type of goods is document

For zone 1

If weight<=500

weight\_first=500

subsequent\_weight=250

additional\_price=0.80

price=4.90

For zone 2

If weight<=500

weight\_first=500

subsequent\_weight=250

additional\_price=1.00

price=5.40

For zone 3

If weight<=500

weight\_first=500

subsequent\_weight=250

additional\_price=1.50

price=6.90

For zone 4

If weight<=500

weight\_first=500

subsequent\_weight=250

additional\_price=1.50

price=7.40

For zone 5

If weight<=500

weight\_first=500

subsequent\_weight=250

additional\_price=2.00

price=7.90

For all zone (zone 1-zone 5)

If weight>500 and weight<=2000

Additional\_weight= weight-weight\_first Weight\_rate=(additional\_weight/subsequent\_weight) Weight\_remainder=(additional\_weight%subsequent\_weight)

If weight\_remainder>0

Weight\_remainder=additional\_price

total\_additional\_price=Math.round((int)weight\_rate)\*additional\_price

total\_price\_zone=total\_additional\_price+weight\_remainder+price

Output

total\_price\_zone\*quantity

Output

Receipt of each zone

If the type of goods is parcel

For zone 1

If weight>2000 and weight <=2500

weight\_first=2500

subsequent\_weight=500

additional\_price=0.50

price=10.50

For zone 2

If weight>2000 and weight <=2500

weight\_first=2500

subsequent\_weight=500

additional\_price=2.00

price=16.00

For zone 3

If weight>2000 and weight <=2500

weight\_first=2500

subsequent\_weight=500

additional\_price=3.00

price=21.00

For zone 4

If weight>2000 and weight <=2500

weight\_first=2500

subsequent\_weight=500

additional\_price=3.50

price=26.00

For zone 5

If weight>2000 and weight <=2500

weight\_first=2500

subsequent\_weight=500

additional\_price=4.00

price=31.00

For all zone (zone 1-zone 5)

If weight>500 and weight<=2000

Additional\_weight= weight-weight\_first Weight\_rate=(additional\_weight/subsequent\_weight) Weight\_remainder=(additional\_weight%subsequent\_weight)

If weight\_remainder>0

Weight\_remainder=additional\_price

total\_additional\_price=Math.round((int)weight\_rate)\*additional\_price

total\_price\_zone=total\_additional\_price+weight\_remainder+price

Output

total\_price\_zone\*quantity

Output

Receipt for all zone

Output

Display “Do you want to place order? 1 is YES,2 is NO”

While user enter 2

Display receipt for all zone

Display “1 Back to Main Menu, 2 Exit”

SWITCH for value menu.

CASE menu is equal to 1

Call method MainMenu.main(args).

Break.

DEFAULT

Call method Exit.main(args).

Break.

END SWITCH

Return main().

Samedaydelivery.main(args).

Set grandLocal=0

Set grandCross=0

Start while (true) loop

Read weight, less than 0 to exit

If weight <0 then

Break loop

Set

grandBoth= grandLocal+grandCross

print grand total of local town + grandLocal

print grand total of cross town + grandCross

print grand total of both town + grandBoth

Get input 1 to go to subprogram main menu, input 2 to go to subprogram exit

Else

Read input to calculate, 1 local, 2 cross town, else invalid

If input =1 then

// Calculate local delivery

If weight >0 and weight < 500 then

Domesticcharge=RM4.90

Surcharge=RM6.00

Total=RM10.90

Printlocalcharge

Else if weight >=500 and <750 then

Domesticcharge=RM5.70

Surcharge=RM6.00

Total=RM11.70

Printlocalcharge

Else if weight >=750 and <=1000 then

Domesticcharge=RM6.50

Surcharge=RM6.00

Total=RM12.50

Printlocalcharge

Else

Print no rates available

End if

Set grandLocal=grandLocal + Total

Else if input=2 then

//Calculate crosstown delivery

If weight >0 and weight < 500 then

Domesticcharge=RM5.40

Surcharge=RM7.50

Total=RM12.90

Printcrosstowncharge

Else if weight >=500 and <750 then

Domesticcharge=RM6.40

Surcharge=RM7.50

Total=RM13.90

Printcrosstowncharge

Else if weight >=750 and <=1000 then

Domesticcharge=RM7.40

Surcharge=RM7.50

Total=RM14.90

Printcrosstowncharge

Else of weight

Print no rates available

End if weight

Else

Print invalid, try again

End if

Set grandCross = grandCross + Total

End of while loop

Printlocalcharge

//Print local receipt

Print local delivery charges

Print local weight+weight

Print local domestic charge+domesticCharge

Print local surcharge+surcharge

Print local total+total

Print local grant total+grant total

End printlocalcharge

Printcrosstowncharge

//Print crosstown receipt

Print crosstown delivery charges

Print crosstown weight+weight

Print crosstown domestic charge+domesticCharge

Print crosstown surcharge+surcharge

Print crosstown total+total

Print crosstown grant total+grant total

End printcrosstowncharge

Return main().

PrepaidBEnvelope(args)

Declare variable for typeDelivery, jenis, price, totalPrice, weight, grandPrice, numberDelivery, menu, number, follow, num and i.

Set size of typeDelivery, price, totalPrice and numberDelivery to 20.

Initialize i to 0.

Initialize grandPrice to 0.

Display Menu of Prepaid Box & Envelope.

DO WHILE follow is not equal to 1 and not equal to 2

Output “Do you want to order(1 to order, 2 to close):”.

Input follow.

END DO

WHILE follow is not equal to 2

DO WHILE weight is more than 10.00 or weight is less than 0

Output “Do you want to order(1 to order, 2 to close):”.

Input weight.

END DO

Call method WeightofDelivery(weight).

jenis = WeightofDelivery(weight).

Print “Your Type of Delievery Recommended”.

Output jenis.

DO WHILE follow is not equal to 1 and not equal to 2

Output “Do you follow it? (1 to Yes / 2 to No):”.

Input follow.

END DO

IF follow is equal to 2 THEN

DO WHILE num not equal to 1 and num is not equal to 2 and num is not equal to 3 and num is not equal to 4 and num is not equal to 5

Output “Enter the type of delivery you want”.

Output “(1.ES / 2.EL / 3.PBS / 4.PBM / 5.PBL):”.

Input num

Call method numForJenis(num)

jenis = numForJenis(num).

END DO

END IF

typeDelivery[i] = jenis.

Output “Enter the number of delivery you want:”

Input number

numberDelivery[i] =number

Call method FindPrice(jenis).

price[i] = FindPrice(jenis).

totalPrice[i] = price[i] \* numberDelivery[i]

totalPrice[i] = Math.round(totalPrice[i] \* 100.0) / 100.0

Display Price and Total Price

Output price [i] and totalPrice[i].

DO WHILE follow is not equal to 1 and not equal to 2

Output “Do you still want to buy?(1 to order, 2 to close):”.

Input follow

END DO

grandPrice = grandPrice + totalPrice[i].

Increase i by 1.

END DO

Display Receipt.

Initialize i to 0.

WHILE price[i] is not equal to 0

Output typeDelivery[i], price[i], numberDelivery[i], totalDelivery[i] and grandPrice[i].

Increase i by 1.

END WHILE

SWITCH for value menu.

CASE menu is equal to 1

Call method MainMenu.main(args).

Break.

DEFAULT

Call method Exit.main(args).

Break.

END SWITCH

WeightofDelivery (double weight)

Declare variable for jenis

IF weight is less than or equal to 0.50 THEN

jenis = “ES”.

ELSE IF weight is more than 0.5 and weight is less than or equal to 1.00 THEN

jenis = “EL”

ELSE IF weight is more than 1.00 and weight is less than or equal to 2.00 THEN

jenis = “PBS”

ELSE IF weight is more than 2.00 and weight is less than or equal to 5.00 THEN

jenis = “PBM”

ELSE THEN

jenis = “PBL”

END IF

Return jenis.

numForJenis (int num)

Declare variable for jenis

IF num is equal to 1 THEN

jenis = “ES”.

ELSE IF num is equal to 2 THEN

jenis = “EL”

ELSE IF num is equal to 3 THEN

jenis = “PBS”

ELSE IF num is equal to 4 THEN

jenis = “PBM”

ELSE THEN

jenis = “PBL”

END IF

Return jenis.

FindPrice(String jenis)

Declare variable for harga.

SWITCH for value jenis

CASE jenis is equal to “ES”

harga = 7.31.

Break.

CASE jenis is equal to “EL”

harga = 10.49.

Break.

CASE jenis is equal to “PBS”

harga = 13.78.

Break.

CASE jenis is equal to “PBM”

harga = 21.20.

Break.

DEFAULT

harga = 31.80.

Break.

END SWITCH.

Return harga.

Return main().

PosEkspres (args).

Declare [][]arr, [][]arr2, [][]arr3, total, row, choice, count, num

do

Call method menu()

Output: Your choice

Input: choice

If choice=0

True = Output: "Thank you and Bye"

break

False = call method count=processChoice(arr,arr2,arr3,total,row,choice, count, num)

while (choice!=0)

End

SWITCH for value menu.

Case 1

call method MainMenu.main(args).

break.

Default

call method Exit.main(args).

break.

END SWITCH

Return main().

method menu()

Output: Menu

1. Add item

2. Display

0. Exit

Return main()

method processChoice()

switch choice()

Case 1

call method count=addItem(arr,arr2,arr3,row,count,num)

break

Case 2

call method display(arr,arr2,arr3,total,row,count)

break

default

Output: "Sorry your choice is not in the list. Please choose again."

Return count

method addItem(arr,arr2,arr3,row,count,num)

if (count < arr.length)

False = Output: "List is FULL."

Return count

True = Output: "Please enter the number that want to pos:"

Input: number\_of\_row

for row < number\_of\_row

False = Return count

True = Output: "Pos " + (count+1) + ": "

arr2[count][3] = count+1

Output: "Please enter the type of pos(LG/LE/LD/LK) :"

Input: arr[count][0]

while(!arr[count][0].equals("LG")&&!arr[count][0].equals("LE")&&!arr[count][0].equals("LD")&&!arr[count][0].equals("LK"))

Output: "Please enter the correct type."

Input: arr[count][0]

switch (arr[count][0])

case "LG"

arr[count][0]="LG";

arr[count][1]="220mmx110mm";

arr2[count][0] =100;

arr2[count][1] =3;

arr3[count][0]=3.18;

Output: "Please enter the quantity for this type: "

Input: arr2[count][2]

Calculate arr3[count][1]=arr3 [count][0]\*arr2 [count][2]

Break

case "LE"

arr[count][0]="LE(C5)";

arr[count][1]="229mmx162mm";

arr2[count][0] =250;

arr2[count][1] =5;

arr3[count][0]=3.71;

Output: "Please enter the quantity for this type: "

Input: arr2[count][2]

Calculate arr3[count][1]=arr3 [count][0]\*arr2 [count][2]

Break

case "LD"

arr[count][0]="LD(B4)";

arr[count][1]="353mmx250mm";

arr2[count][0] =500;

arr2[count][1] =10;

arr3[count][0]=4.77;

Output: "Please enter the quantity for this type: "

Input: arr2[count][2]

Calculate arr3[count][1]=arr3 [count][0]\*arr2 [count][2]

Break

case "LK"

arr[count][0]="LK";

arr[count][1]="340mmx250mm";

arr2[count][0] =1000;

arr2[count][1] =25;

arr3[count][0]=7.42;

Output: "Please enter the quantity for this type: "

Input: arr2[count][2]

Calculate arr3[count][1]=arr3 [count][0]\*arr2 [count][2]

Break

count ++

row ++

method display(arr,arr2,arr3,total,row,count)

if (count==0)

True = Output: "Sorry list is empty."

False = "Pos Express" "NO","TYPE ","SIZE ","WEIGHT(GM) ","THICKNESS (MAX)","UNIT PRICE(RM)","Quantity","AMOUNT"

for row < count

True = calculate total+=arr3[row][1]

row++

False = Output: "TOTAL DUE: "

Return display()

Exit.main(args)

Declare variable grand[i] and grandTotal.

grand = {Poslaju.grand(), Samedaydelivery.grand(), PrepaidBEnvelope.grand(),PosEkspres.grand()}

Initialize grandTotal to 0.

Display Receipt

Output grand [0], grand [1], grand[2] and grand[3].

For (int i = 0; i < grand.length; i++)

grandTotal = grandTotal + grand[i];

End For

Output grandTotal.

Print “Thank You, Bye”.

Return main()

End