import module

class product

defines product with an undefined name variable

class pizza that is a child class of product

contains name variable from product

contains undefined variables crust, sauce, cheese, and toppings

class main

customer\_order := {}

costs := []

subtotal := 0

tax := 0.275

delivery fee := 3.99

prints welcome to domino’s

order := “n”

yesorno := “y” or “n” based on if user wants to order

yesorno := call: check valid

IF yesorno equals “y” THEN

order := “y”

ELSE

Print have a nice day

ENDIF

WHILE order equals “y”

FOR product in menu

i := 1

productyn := would you like to order this product

productyn := call: check valid

IF productyn equals “y”

pizza counter := 1

IF product equals pizza in menu

Continue ordering = “y”

WHILE continue ordering equals “y”

inp := would you like premade or custom

inp := call: check valid

IF inp equals 1

Call: pizza menu

choice := which pizza

choice := call: check valid

amount := how many

amount := call: check valid

customer\_order := choice equals amount

costs add cost x amount

subtotal := amount x cost

ELSE

amount := how many pizza

amount := call: check valid

FOR p in amount

obj := product equals pizza + pizza counter

call: crust menu

crust := which crust

crust := call: check valid

call: sauce menu

sauce:= which sauce

sauce := call: check valid

call: cheese menu

cheese := which cheese

cheese := call: check valid

call: toppings menu

toppings := which toppings

topings := call: check valid

pizza counter = pizza counter + 1

customer\_order = name (details) equals 1

costs add cost x amount

subtotal := amount x cost

continue ordering := continue order

continue ordering := call: check valid

ELSE

Get product in menu

continue ordering := “y”

WHILE continue ordering equals “y”

choice := which product

choice := call: check valid

amount := how many

amount := call: check valid

customer\_order := choice equals amount

costs add cost x amount

subtotal := amount x cost

continue ordering := continue order

continue ordering := call: check valid

i := Ii+ 1

order := “n”

call: delivery date

subtotal := round subtotal

after tax := round subtotal x tax

Prints text that looks like a receipt with all the costs and details from user’s order