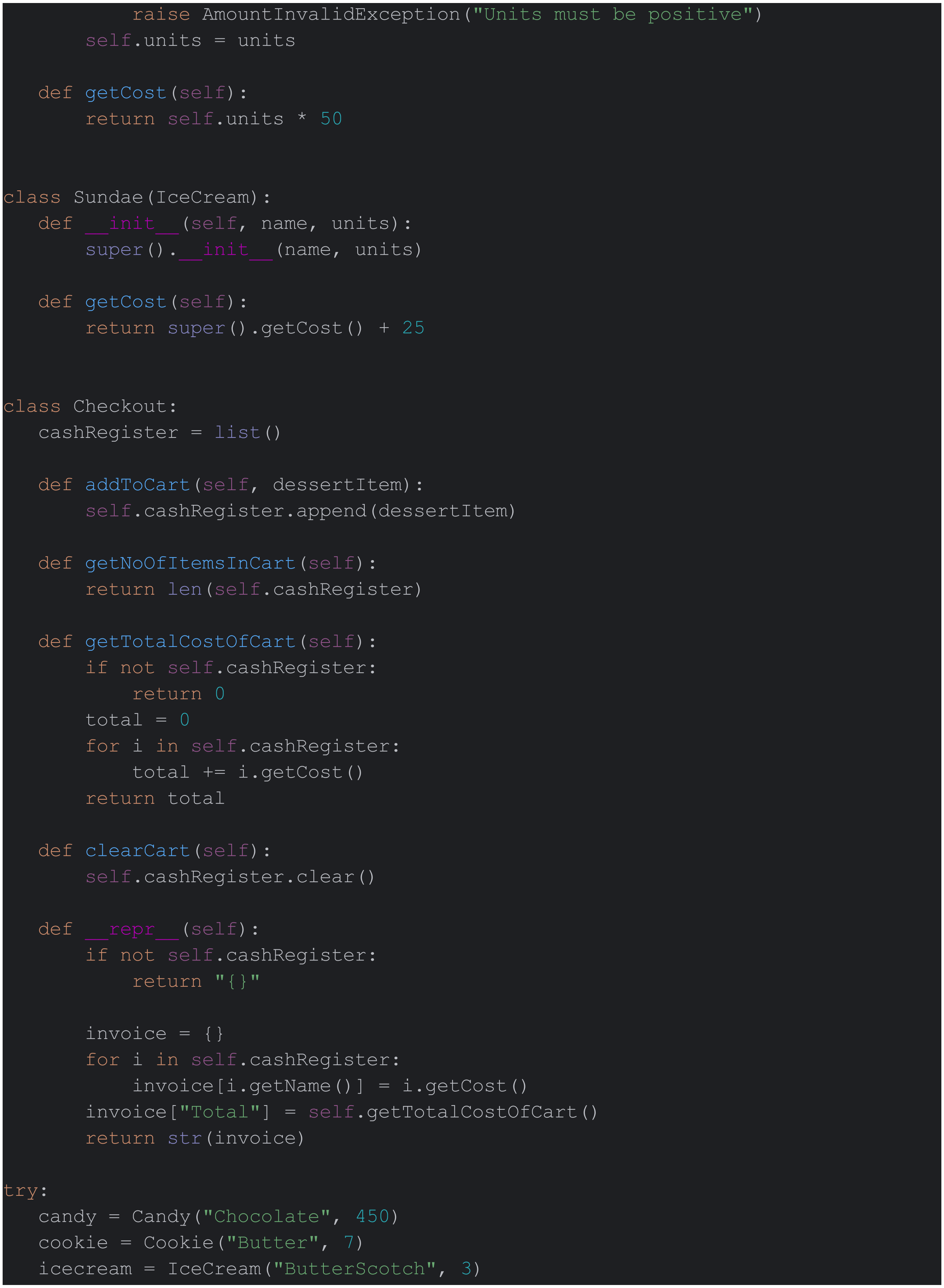
250240128002 Aditya Bhawsar

```
from abc import ABC, abstractmethod

class CartIsEmptyException(Exception):
   pass

class AmountInvalidException(Exception):
   pass

class DesertItem(ABC):
   def __init__(self, name):
       self.name = name

   @abstractmethod
   def getCost(self):
       pass

   def getName(self):
       return str(self.name)


class Candy(DesertItem):
   def __init__(self, name, weightInGms):
       super().__init__(name)
       if weightInGms < 0:
           raise AmountInvalidException("Weight must be positive")
       self.weightInGms = weightInGms

   def getCost(self):
       return (self.weightInGms * (50 / 1000))


class Cookie(DesertItem):
   def __init__(self, name, units):
       super().__init__(name)
       if units < 0:
           raise AmountInvalidException("Units must be positive")
       self.units = units

   def getCost(self):
       return (self.units * (120 / 12))


class IceCream(DesertItem):
   def __init__(self, name, units):
       super().__init__(name)
       if units < 0:
```



```
           raise AmountInvalidException("Units must be positive")
       self.units = units

   def getCost(self):
       return self.units * 50


class Sundae(IceCream):
   def __init__(self, name, units):
       super().__init__(name, units)

   def getCost(self):
       return super().getCost() + 25


class Checkout:
   cashRegister = list()

   def addToCart(self, dessertItem):
       self.cashRegister.append(dessertItem)

   def getNoOfItemsInCart(self):
       return len(self.cashRegister)

   def getTotalCostOfCart(self):
       if not self.cashRegister:
           return 0
       total = 0
       for i in self.cashRegister:
           total += i.getCost()
       return total

   def clearCart(self):
       self.cashRegister.clear()

   def __repr__(self):
       if not self.cashRegister:
           return "{}"

       invoice = {}
       for i in self.cashRegister:
           invoice[i.getName()] = i.getCost()
       invoice["Total"] = self.getTotalCostOfCart()
       return str(invoice)

try:
   candy = Candy("Chocolate", 450)
   cookie = Cookie("Butter", 7)
   icecream = IceCream("ButterScotch", 3)
   sundae = Sundae("Butterscotch large", 2)

   checkout = Checkout()

   checkout.addToCart(candy)
   checkout.addToCart(cookie)
   checkout.addToCart(sundae)
   checkout.addToCart(icecream)

   print(checkout)
   checkout.clearCart()
   print(checkout)

   if not checkout.cashRegister:
       raise CartIsEmptyException("Cart is empty.")

except CartIsEmptyException as e:
   print(e)
except AmountInvalidException as e:
   print("Error: ",e)
except Exception as e:
   print("An unexpected error occurred: ",e)
```