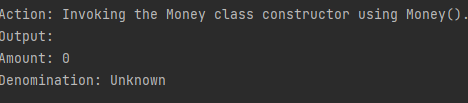
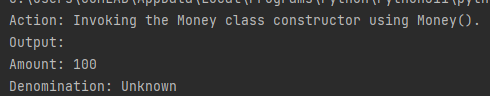
Problem 1:

class Money:  
  
 def \_\_init\_\_(self, amount: int = 0, denomination: str = "Unknown"):  
 self.amount = amount  
 self.denomination = denomination  
  
 def \_\_str\_\_(self):  
 return (f"Action: Invoking the Money class constructor using Money().\n"  
 f"Output:\n"  
 f"Amount: {self.amount}\n"  
 f"Denomination: {self.denomination}")  
  
def test\_money():  
 t = Money(0,"Unknown")  
 print(t)  
   
if \_\_name\_\_ == '\_\_main\_\_':  
 test\_money()

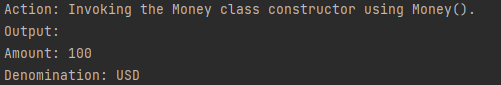
Sample 1:



Sample 2:



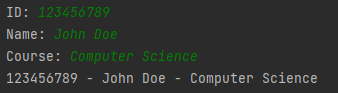
Sample 3:



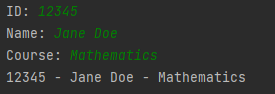
Problem 2:

class Student:  
  
 def \_\_init\_\_(self, id: int = 0, name: str = "Unknown", course: str = "Unknown", length: str = 0):  
 self.id = id  
 self.name = name  
 self.course = course  
 self.length = length  
  
 def \_\_str\_\_(self):  
 if self.id == str:  
 self.validate\_info()  
 elif self.length > 9:  
 self.validate\_info()  
 elif self.name == int:  
 self.validate\_info()  
 elif self.course == int:  
 self.validate\_info()  
 else:  
 return (f"{self.id} - {self.name} - {self.course}")  
  
 def validate\_info(self):  
 print("Student information is not valid.")  
  
def test\_student():  
 id = int(input("ID: "))  
 name = input("Name: ")  
 course = input("Course: ")  
 length = len(str(id))  
 stud = Student(id,name,course,length)  
 print(stud)  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 test\_student()

Sample 1:



Sample 2:



Sample 3:

