1

```
with open(r"C:\Users\zkayd\Desktop\py\1\charles.txt", "r") as file:
    text = file.read()

lines = text.split("\n")
words = text.lower().replace(".", "").replace(",", "").split()
word_count = {}
for word in words:
    if word in word_count:
        word_count[word] += 1
    else:
        word_count[word] = 1
with open("analysis.txt", "w") as file:
    file.write("Lines: " + str(len(lines)) + "\n")
    file.write("Words: " + str(len(words)) + "\n")
    file.write("Frequency: " + "\n")
    for word in word_count:
        file.write(word + ": " + str(word_count[word]) + "\n")
```

2

```
import json

with open("22.json", "r", encoding="utf-8") as file:
    students = json.load(file)
for student in students:
    grades = student["grades"]
    student["average_grade"] = sum(grades) / len(grades)
with open("students_updated.json", "w", encoding="utf-8") as file:
    json.dump(students, file, indent=4)
```

3

```
class Person:
    def __init__(self, name):
        self._name = name

    def get_info(self):
        return f"Name: {self._name}"
class Student(Person):
    def __init__(self, name, student_id):
        super().__init__(name)
        self.student_id = student_id
    def get_info(self):
        return f"Student Name: {self._name}, ID: {self.student_id}"
person = Person("Ivan")
student = Student("Zarina", "1633")
print(person.get_info())
print(student.get_info())
```

4

```
class Employee:
    def __init__(self, salary):
        self._salary = salary

    def get_salary(self):
        return self._salary
    def get_role(self):
        return "Employee"
class Manager(Employee):
    def __init__(self, salary, bonus):
        super().__init__(salary)
        self.bonus = bonus
    def get_role(self):
        return "Manager"
    def get_bonus(self):
        return self.bonus
def print_employee_info(employees):
    for emp in employees:
        print(f"Role: {emp.get_role()}, Salary: {emp.get_salary()}")
emp1 = Employee(5700)
emp2 = Manager(3900, 1600)
print_employee_info([emp1, emp2])
```

5

```
class BankAccount:
    def __init__(self, owner, balance=0):
        self.__owner = owner
        self.__balance = balance

    def deposit(self, amount):
        if amount <= 0:
            print("Deposit must be positive")
        else:
            self.__balance += amount
    def withdraw(self, amount):
        if amount > self.__balance:
            print("Insufficient balance")
        elif amount <= 0:
            print("Withdrawal must be positive")
        else:
            self.__balance -= amount
    def get_balance(self):
        return self.__balance
account = BankAccount("Zarina", 2500)
account.deposit(400)
account.withdraw(800)
print(account.get_balance())
```