# Task: Create a Hierarchy of Classes for Managing Electronic Devices

## 1. Create an abstract class `ElectronicDevice` with the following members:

- Abstract methods:  
 - `GetPowerConsumption(): double` - returns the power consumption of the device.  
 - `GetBrand(): string` - returns the brand of the device.  
- Virtual methods:  
 - `Operate(): string` - this method should return a message in the format:  
 `$"Operating a {classType.Name}"`, where `classType.Name` is the name of the class.

## 2. Derive the class `ElectronicDevice` into two subclasses:

- `Laptop`  
- `Smartphone`

## 3. For each of them:

- Add fields:  
 - For `Laptop`: `ramSize` (amount of RAM) and `processorType` (type of processor).  
 - For `Smartphone`: `batteryCapacity` (battery capacity in mAh) and `screenSize` (screen size in inches).  
- Provide encapsulation for these fields.  
- Add a public constructor to initialize all fields.  
- Implement specific methods for `GetPowerConsumption` and `GetBrand`.  
- Override the method `Operate` to return an appropriate message for the respective device type.

## 4. Demonstrate polymorphism:

- Create an array of objects of type `ElectronicDevice`, containing both laptops and smartphones.  
- Iterate through the array and invoke the methods `Operate`, `GetPowerConsumption`, and `GetBrand` for each object.