🔹 **Statement:**

"Abstraction allows making relevant information visible and encapsulation enables a programmer to implement the desired level of abstraction."

✅ **Meaning:**

This statement connects **abstraction** and **encapsulation** — both are core principles of **Object-Oriented Programming (OOP)**.

🔶 **1. Abstraction – *What it does?***

- Shows **only the necessary details** to the user.
- **Hides unnecessary internal code** or logic.
- Helps the user focus on **what the object does**, not *how* it does it.

Example:
When you drive a car:

- You use the steering, brake, and accelerator.
- You don't need to know how the engine or brake system works.

🔶 **2. Encapsulation – *How abstraction is achieved?***

- Encapsulation is the process of **binding data + methods** together inside a class.
- It **hides the internal details** using access modifiers (private, protected, etc.).
- This allows the programmer to control **how much information is visible** outside the class.

Example:

```
class BankAccount {
  private double balance;  // internal detail hidden


  public void deposit(double amount) {
    balance += amount;   // abstraction: only deposit method is shown
  }
}
```

- balance is hidden (encapsulation).
- User interacts with only deposit() (abstraction).

🔹 **So, what does the original line mean?**

- **Abstraction** defines *what* the user should see (relevant details).
- **Encapsulation** controls *how* to show only that part and hide the rest.

🔁 In short:
🟡 *Abstraction* = "Show only what's necessary"
🟢 *Encapsulation* = "Hide the rest and protect the data"