***PHASE 4***

*SMART WATER FOUNTAIN*

*SUNMATHI.E*

*PPG INSTITUTE OF TECHNOLOGY*

*COIMBATORE*

***PROGRAM:***

*Public class SmartWaterFountain*

*{*

*Private boolean isOn;*

*Private int waterLevel;*

*Public SmartWaterFountain()*

*{*

*isOn = false;*

*waterLevel = 0;*

*}*

*Public void turnOn()*

*{*

*isOn = true;*

*System.out.println(“Fountain is now on.”);*

*}*

*Public void turnOff()*

*{*

*isOn = false;*

*System.out.println(“Fountain is now off.”);*

*}*

*Public void fillWater(int amount)*

*{*

*If (isOn)*

*{*

*waterLevel += amount;*

*System.out.println(“Water level increased by “ + amount+ liters.”);*

*}*

*Else*

*{*

*System.out.println(“Cannot fill water when the fountain is off.”);*

*}*

*}*

*Public void dispenseWater(int amount)*

*{*

*If (isOn)*

*{*

*If (waterLevel >= amount)*

*{*

*waterLevel -= amount;*

*System.out.println(“Dispensing “ + amount + “ liters of water.”);*

*}*

*Else*

*{*

*System.out.println(“Insufficient water for dispensing.”);*

*}*

*}*

*Else*

*{*

*System.out.println(“Cannot dispense water when the fountain is off.”);*

*}*

*Else*

*Public int getWaterLevel()*

*{*

*Return waterLevel;*

*}*

*Public boolean isOn()*

*{*

*Return isOn;*

*}*

*Public static void main(String[] args)*

*{*

*SmartWaterFountain fountain = new SmartWaterFountain(); Fountain.turnOn();*

*Fountain.fillWater(20);*

*Fountain.dispenseWater(10);*

*System.out.println(“Current water level: “ +fountain.getWaterLevel());*

*Fountain.turnOff();*

*}*

*}*

***OUTPUT:***

*Fountain is now on.*

*Water level increased by 20 liters*

*Dispensing 10 liters of water*

*Current water level: 10*

*Fountain is now off.*

***CONCLUSION:***

1. *You can turn it on or off using the*

*turnOn and `turnoff’ methods.*

1. *The `fillWater method allows adding*

*Water if it’s on.*

1. *The dispenseWater ` method*

*Dispenses water if there’s enough.*

1. *It provides access to the current water*

*Level and on/off status.*

1. *The constructor initializes the fountain*

*As off with no water.*

1. *It includes error messages for improper*

*Actions.*

1. *Demonstrated in the ‘main’ method*

*With on/off, filling, and dispensing.*

1. *Demonstrated in the ‘main method*

*With on/off, filling, and dispensing*

1. *A simple, encapsulated representation*

*Of a smart water fountain.*

1. *Offers essential functionality for*

*Managing water levels.*