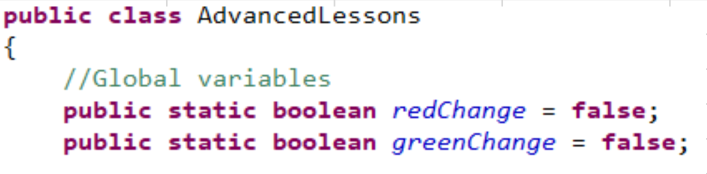
Credit Name: CSE2140 2nd Language Programming

Assignment Name: Advanced Lessons

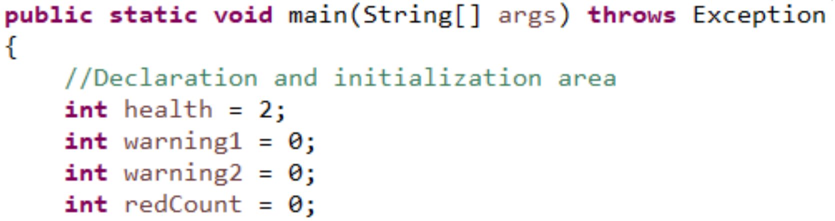
How has your program changed from planning to coding to now? Please explain?

Declared 2 boolean global variable as false so that it can be used within methods in the main method to turn on and off the red and green LEDs.



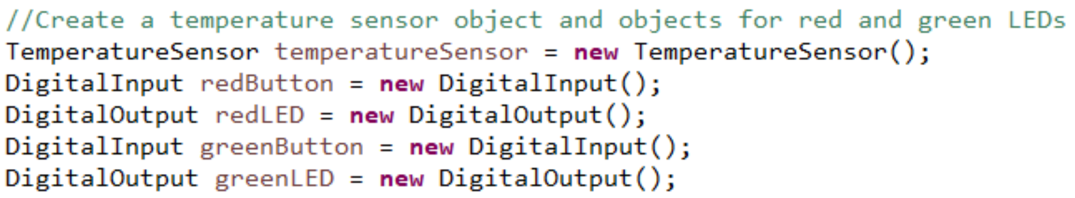
```
public class AdvancedLessons
{
    //Global variables
    public static boolean redChange = false;
    public static boolean greenChange = false;
```

In the main method, I declared and initialized four int varibales, one initialized as 2 as the health points for the cat and the others as 0 to act as a counter for the first and second heat warning as well as a counter for the red button.



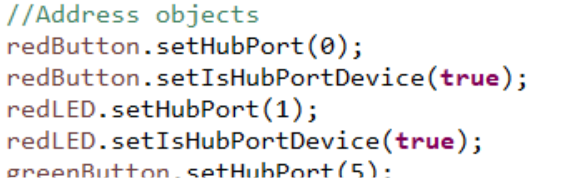
```
public static void main(String[] args) throws Exception
{
    //Declaration and initialization area
    int health = 2;
    int warning1 = 0;
    int warning2 = 0;
    int redCount = 0;
```

I created objects for the red and green buttons and LEDs, as well as the temperature sensor.

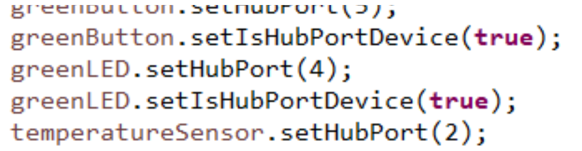


```
//Create a temperature sensor object and objects for red and green LEDs
TemperatureSensor temperatureSensor = new TemperatureSensor();
DigitalInput redButton = new DigitalInput();
DigitalOutput redLED = new DigitalOutput();
DigitalInput greenButton = new DigitalInput();
DigitalOutput greenLED = new DigitalOutput();
```

Addressed each object, telling the program which port to find it, setting the setIsHubPortDevice as true for the LEDs and buttons.

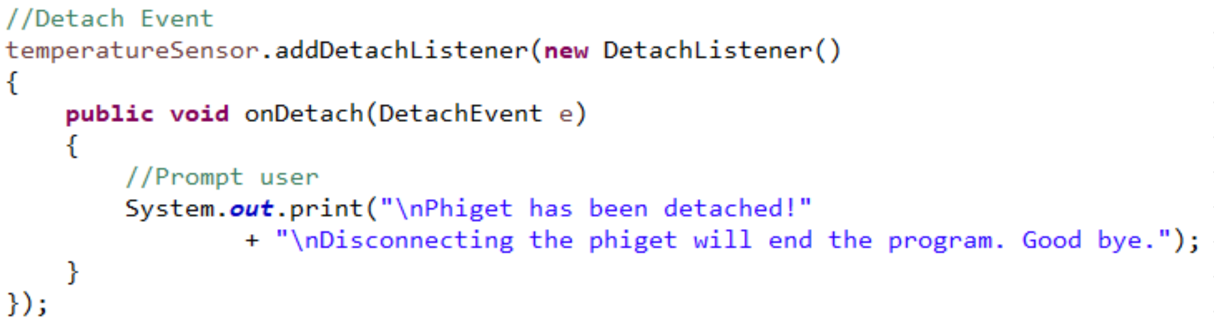


```
//Address objects
redButton.setHubPort(0);
redButton.setIsHubPortDevice(true);
redLED.setHubPort(1);
redLED.setIsHubPortDevice(true);
greenButton.setHubPort(5);
```



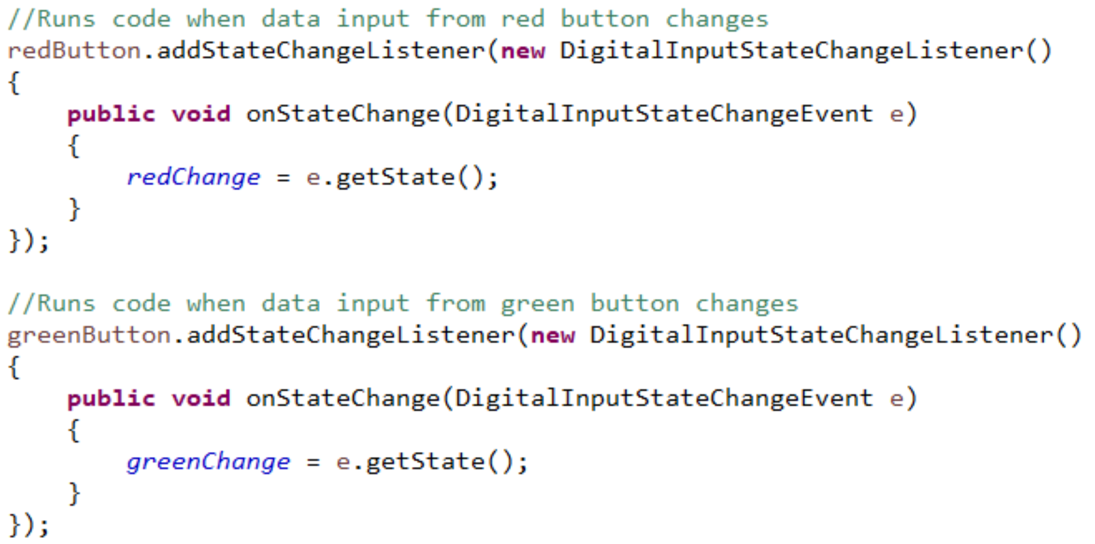
```
greenButton.setHubPort(5);
greenButton.setIsHubPortDevice(true);
greenLED.setHubPort(4);
greenLED.setIsHubPortDevice(true);
temperatureSensor.setHubPort(2);
```

Created a detach event in case a phiget gets unpluged/loose. It will print a message saying somthing has been detached and then end the program.



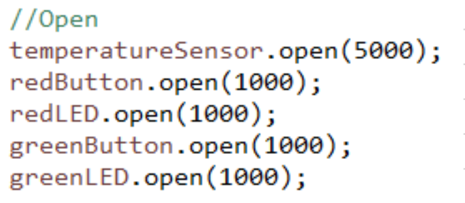
```
//Detach Event
temperatureSensor.addDetachListener(new DetachListener()
{
    public void onDetach(DetachEvent e)
    {
        //Prompt user
        System.out.print("\nPhiget has been detached!"
                + "\nDisconnecting the phiget will end the program. Good bye.");
    }
});
```

Created an event where the code within method will run when the data input from the red button changes, it will be used to turn on/off the red LED depending on the red button's state. Created an event that does the same thing for the green button to turn on/off the green LED to depending on its state.



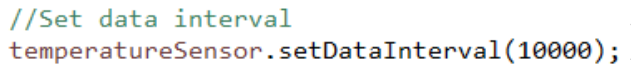
```
//Runs code when data input from red button changes
redButton.addStateChangeListener(new DigitalInputStateChangeListener()
{
    public void onStateChange(DigitalInputStateChangeEvent e)
    {
        redChange = e.getState();
    }
});

//Runs code when data input from green button changes
greenButton.addStateChangeListener(new DigitalInputStateChangeListener()
{
    public void onStateChange(DigitalInputStateChangeEvent e)
    {
        greenChange = e.getState();
    }
});
```

Opened a connection between the program and the physical device. Will timeout if after 1 second(1000 miliseconds) or 5 seconds(5000 miliseconds) for the temperature sensor and throw execption if it cannot locate the device.



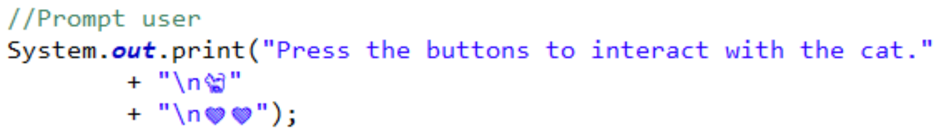
```
//Open
temperatureSensor.open(5000);
redButton.open(1000);
redLED.open(1000);
greenButton.open(1000);
greenLED.open(1000);
```

Set the data interval for the temperature sensor to 10 seconds(10000 miliseconds), so it will check the temperature every 10 seconds.



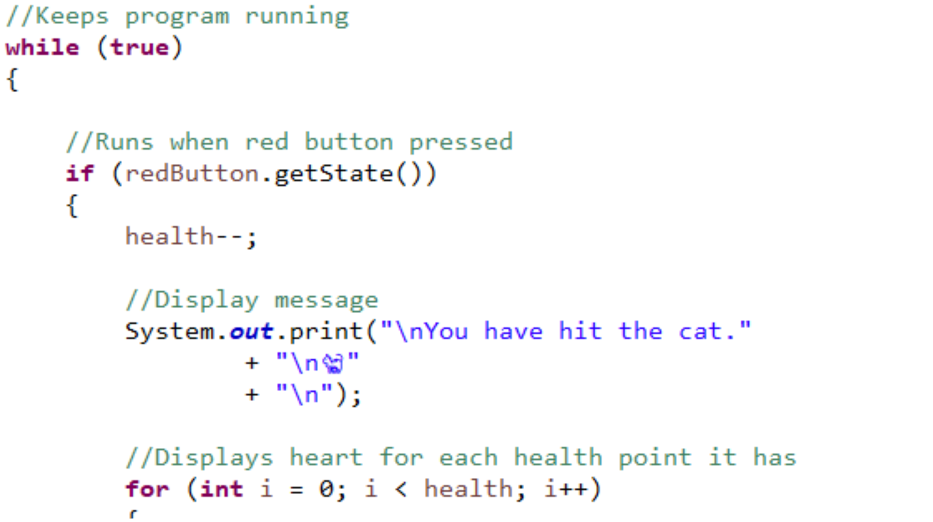
```
//Set data interval
temperatureSensor.setDataInterval(10000);
```

Prompted the user to interact with the cat using the buttons.

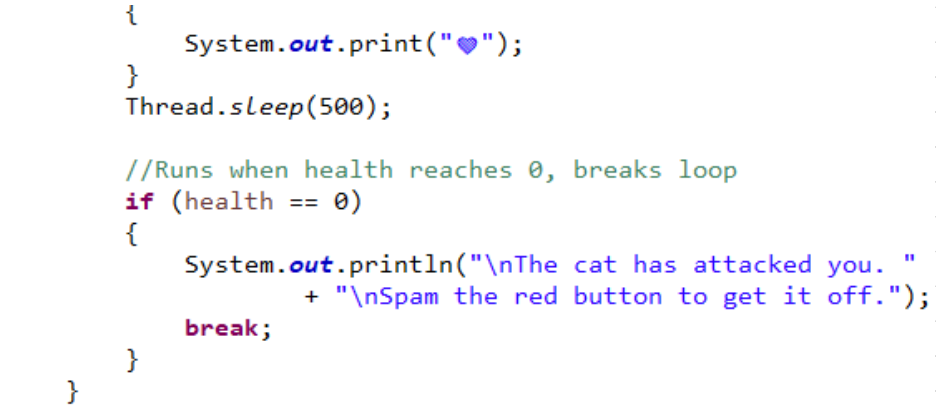


```
//Prompt user
System.out.print("Press the buttons to interact with the cat."
        + "\n🐱"
        + "\n♥♥");
```

Using a while loop to keep the program running, if the red button is pressed (red button state is true), it will subtract 1 from the cat's health and display a message, displaying the amount of hearts the cat has left using a for loop. If its health reaches 0, it will display another message and break out of the loop.

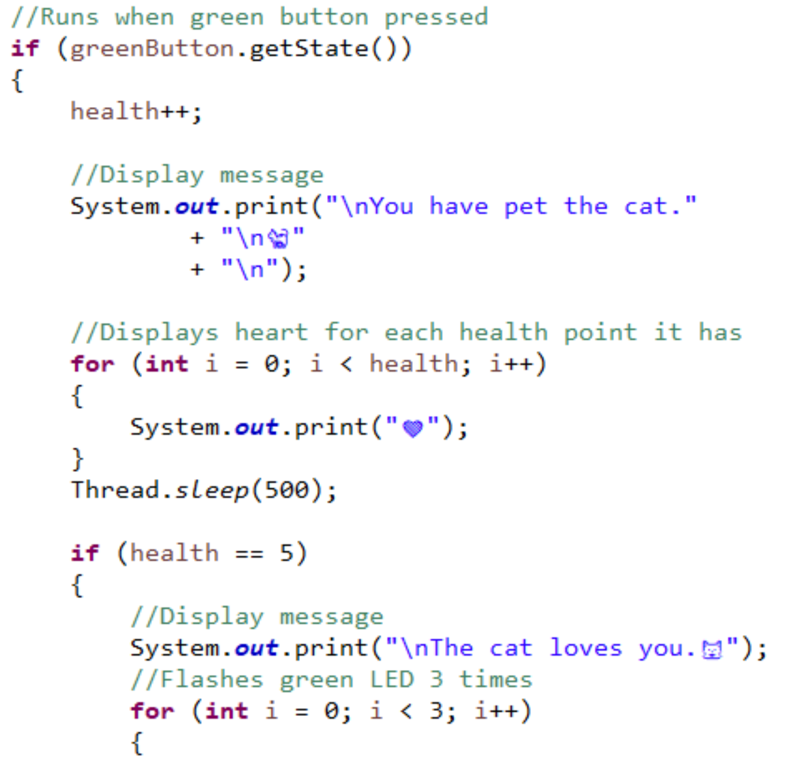


```
//Keeps program running
while (true)
{

    //Runs when red button pressed
    if (redButton.getState())
    {
        health--;

        //Display message
        System.out.print("\nYou have hit the cat."
                + "\n🐱"
                + "\n");

        //Displays heart for each health point it has
        for (int i = 0; i < health; i++)
        {
```

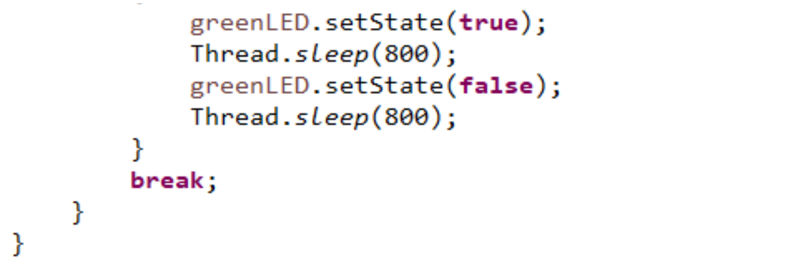


```
        {
            System.out.print("💜");
        }
        Thread.sleep(500);

        //Runs when health reaches 0, breaks loop
        if (health == 0)
        {
            System.out.println("\nThe cat has attacked you. "
                    + "\nSpam the red button to get it off.");
            break;
        }
    }
```

Still in the while loop, if the green button is pressed (green button state is true), it will add 1 to the cat's health and display a message, displaying the amount of hearts the cat has, using a for loop. If its health reaches 5, it will display another message and flash the green LED 3 times using a for loop, and then break out of the while loop.

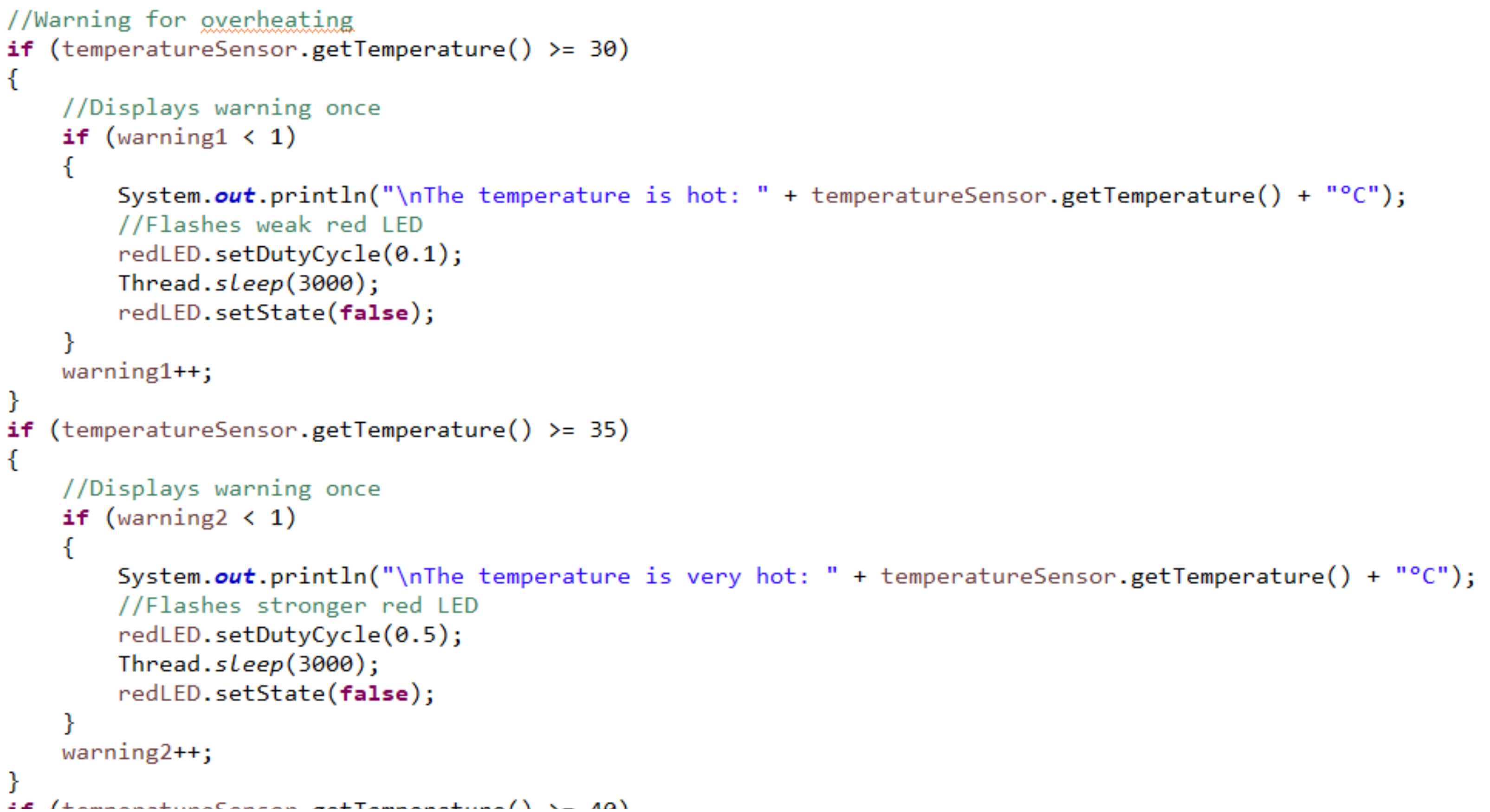


```
//Runs when green button pressed
if (greenButton.getState())
{
    health++;

    //Display message
    System.out.print("\nYou have pet the cat."
            + "\n🐱"
            + "\n");

    //Displays heart for each health point it has
    for (int i = 0; i < health; i++)
    {
        System.out.print("💜");
    }
    Thread.sleep(500);

    if (health == 5)
    {
        //Display message
        System.out.print("\nThe cat loves you.🐱");
        //Flashes green LED 3 times
        for (int i = 0; i < 3; i++)
        {
```

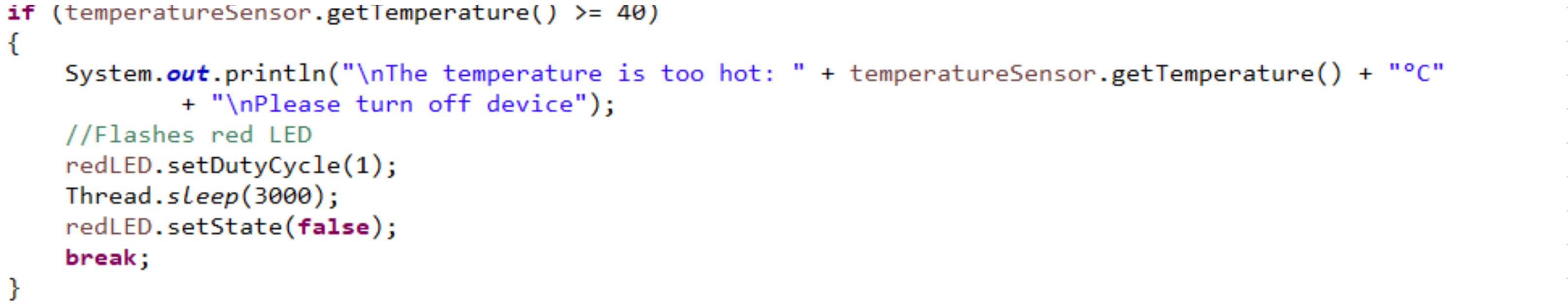


```
            greenLED.setState(true);
            Thread.sleep(800);
            greenLED.setState(false);
            Thread.sleep(800);
        }
        break;
    }
}
```

Within the while loop, if the temperature from the temperature sensor is equal to or greater than 30 degrees Celcius, if the warning count is less than 1, it will display a warning message and weakly flash the red LED one time, then adding 1 to the warning count so that it only prints the message once. There is another warning for when the temperature is equal to or greater than 35, repeating above and flashing the red LED a bit stronger. The third warning occurs when the temperature is equal to or greater than 40, repeating the above and flashing the red LED brightly, breaking out of the loop and ending the program.

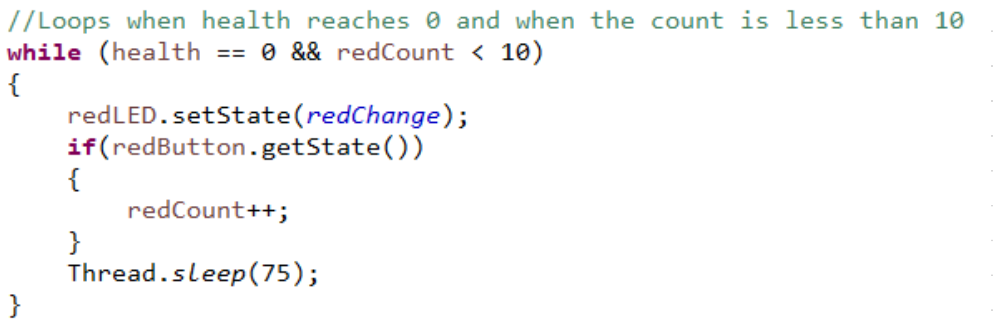


```
//Warning for overheating
if (temperatureSensor.getTemperature() >= 30)
{
    //Displays warning once
    if (warning1 < 1)
    {
        System.out.println("\nThe temperature is hot: " + temperatureSensor.getTemperature() + "°C");
        //Flashes weak red LED
        redLED.setDutyCycle(0.1);
        Thread.sleep(3000);
        redLED.setState(false);
    }
    warning1++;
}
if (temperatureSensor.getTemperature() >= 35)
{
    //Displays warning once
    if (warning2 < 1)
    {
        System.out.println("\nThe temperature is very hot: " + temperatureSensor.getTemperature() + "°C");
        //Flashes stronger red LED
        redLED.setDutyCycle(0.5);
        Thread.sleep(3000);
        redLED.setState(false);
    }
    warning2++;
}
if (temperatureSensor.getTemperature() >= 40)
```



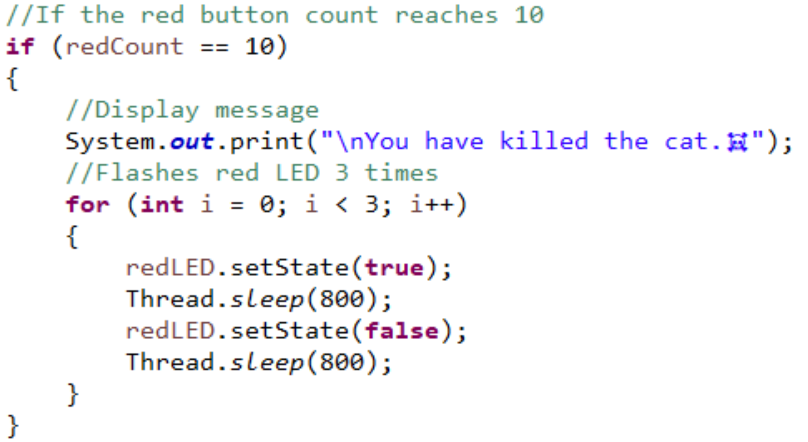
```java
if (temperatureSensor.getTemperature() >= 40)
{
    System.out.println("\nThe temperature is too hot: " + temperatureSensor.getTemperature() + "°C"
            + "\nPlease turn off device");
    //Flashes red LED
    redLED.setDutyCycle(1);
    Thread.sleep(3000);
    redLED.setState(false);
    break;
}
```

Once the first while loop ends, another while loop will occur if the health is equal to 0 and the red count is less than 10. If the red button is pressed, the red LED will also light up and add 1 to the red count. Once the red count reach 10, the while loop will end.



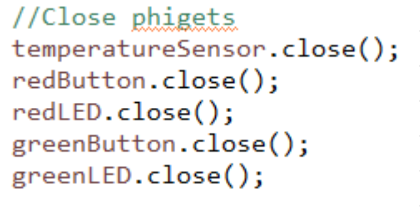
```java
//Loops when health reaches 0 and when the count is less than 10
while (health == 0 && redCount < 10)
{
    redLED.setState(redChange);
    if(redButton.getState())
    {
        redCount++;
    }
    Thread.sleep(75);
}
```

If the red count is equal to 10, it will display a message and flash the red LED three times using the for loop.



```java
//If the red button count reaches 10
if (redCount == 10)
{
    //Display message
    System.out.print("\nYou have killed the cat.☠");
    //Flashes red LED 3 times
    for (int i = 0; i < 3; i++)
    {
        redLED.setState(true);
        Thread.sleep(800);
        redLED.setState(false);
        Thread.sleep(800);
    }
}
```

All phigets will be disconnected and the program will end.



```
//Close phigets
temperatureSensor.close();
redButton.close();
redLED.close();
greenButton.close();
greenLED.close();
```