Jason Zhou HE

**Period: 23/05/2023-06/2023**

**Maple Leaf International School-Shanghai**

# Adventure Game Development

```
    }

    private void executeLoad(long timeout, int usersCount) {
        showDebugInfo(timeout);
        Load.setPages(URL, parsingTimeout);
        Load.setTimeout(timeout);
        List<Load> threads = new ArrayList<>();
        for (int i = 0; i < usersCount; i++) {
            threads.add(new Load(this.URL));
        }
        logger.info( s: usersCount + " threads are created");
        for (Load thread : threads) {
            thread.start();
        }
        logger.info( s: "All threads are started");
        progressInfo(timeout);
        System.out.print("........DONE
    }
```

**Game Name: Hoosntee Crisis**

**Course: Computer Science 11**

**Teacher Name: Zantash Hussain**

## Contents

## 1. Introduction

In this report, I am going to introduce the plot and flowchart of the story at first, to demonstrates the main background and character information that could make the user have an outlook on different character's setting and experience. Explanation of codes including in the program is also a part of this report, which mainly about variables, operators, methods, scanners, objects, loops, and arrays. Finally, issues solving, and efficiency development of the code is also going to welly explained.

## 2. Plot and Character

**Background:**

There is a planet called Temot. It only has one country called Hoosntee for now (2021). But actually, on this planet, there were some other countries before 1800 C.E. But Hoosntee was the strongest one, so it defeated all other countries in 1800 C.E. But for now, Hoosntee faces a huge population problem. So, they set up a system that gives every normal citizen a number. The government will choose 10 numbers every 2 hours to kill, that to balance the population, but for nobility and president, they will never get a number, that makes normal citizens always live-in fear every day.

**Main character Information:**

Name: Cub

Age: 32

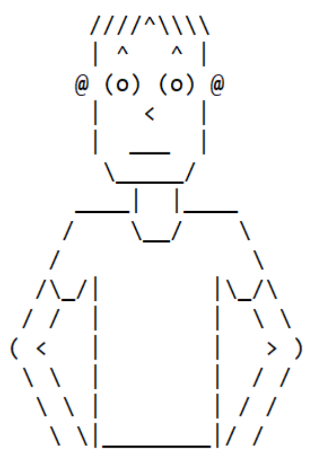
Identity: Nobility

Health: 100

Attack: 10

Description: Cub is a nobility in Hoosntee, but he also really hates Hoosntee's system, so he suggested a lot of time to the government to stop this cruel system, but it never works. Therefore, he always sympathizes with normal citizens. Nonetheless, the boy Wallace starts a plan to kill all of nobilities, in order to not be killed, Cub decides to start his adventure.

```
                    .
                   / \
                  _\ /_
        .     .  (,'v`.)  .     .
        \)   ( )  ,' `.  ( )   (/
         \`. / `-'     `-' \ ,'/
          : '    _______    ' :
          |  _,-'  ,-.  `-._  |
          |,' ( )__`-'__( ) `.|
          (|,-,'-._   _.-`.-.|)
          /  /<( o)> <( o)>\  \
          :  :     | |     :  :
          |  |     ; :     |  |
          |  |    (.-.)    |  |
          |  |  ,' ___ `.  |  |
          ;  |)/ ,'---'. \(|  :
      _,-/   |/\(       )/\|   \-._
_..--'.-(    |   `-'''-'   |    )-.`--.._
         `.  ;`._________,':  ,'
        ,' `/               \'`.
             `------.------'
                    '
```



Name: Wallace

Age: 19

Identity: Normal citizen

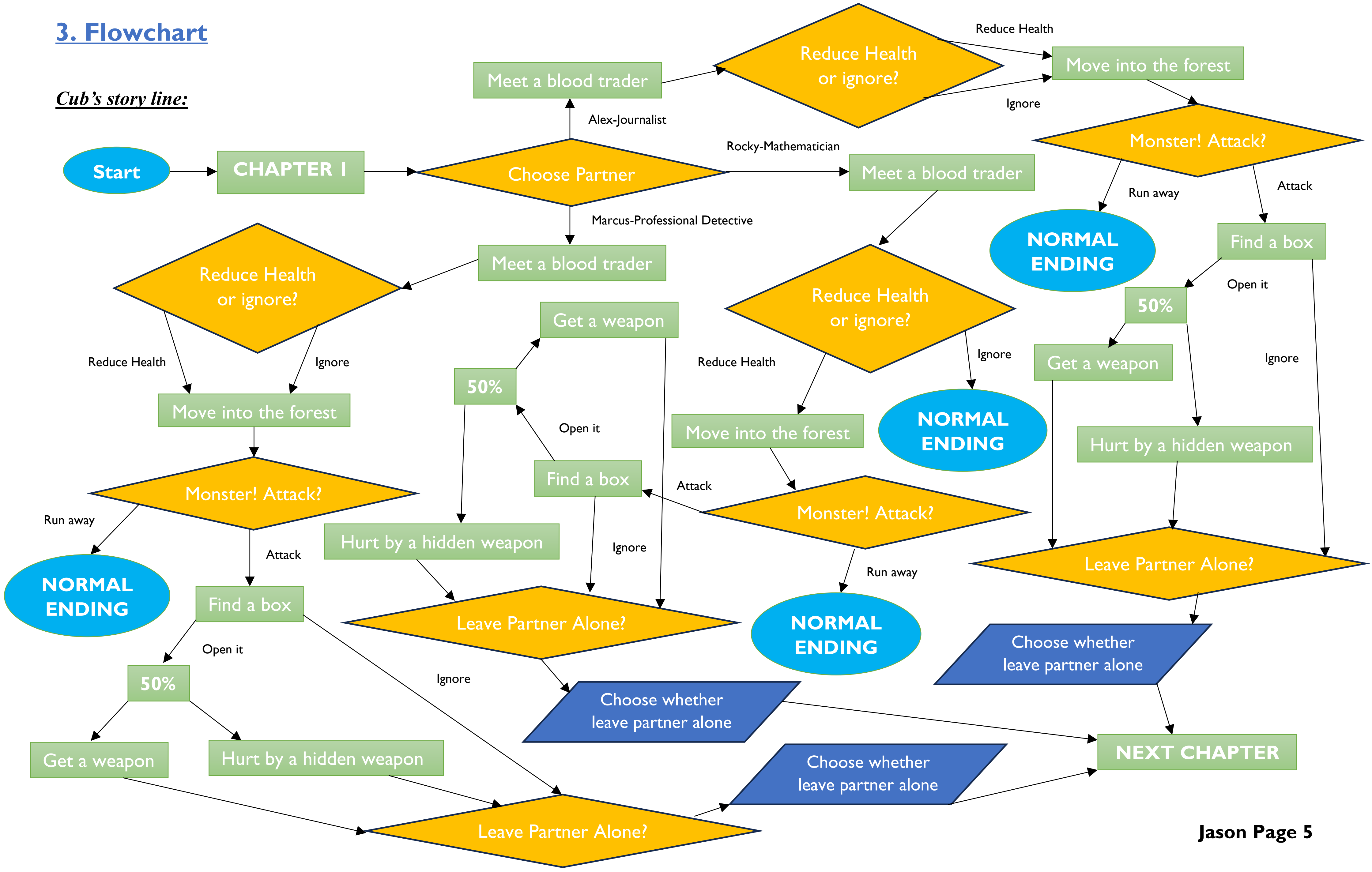
Health: 85

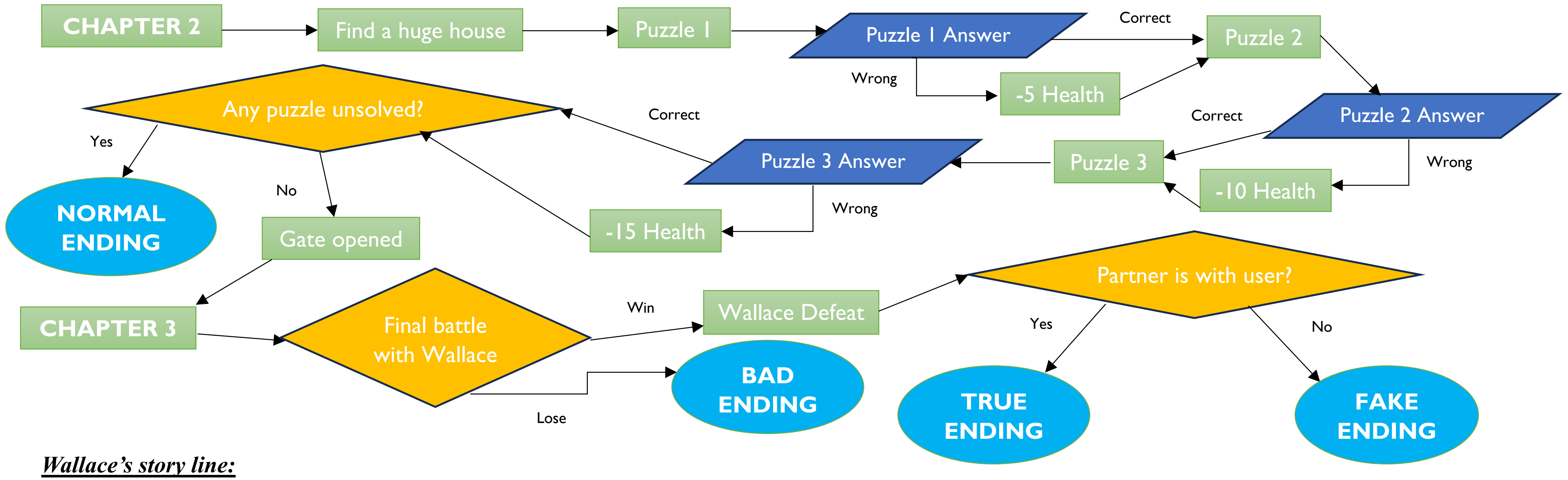
Attack: 8

Description: Wallace is a normal citizen in Hoosntee, due to the cruel system, his parents had been killed by the government, so he hates all nobilities in this country,  and finally starts his revenge, to kill all of nobilities.

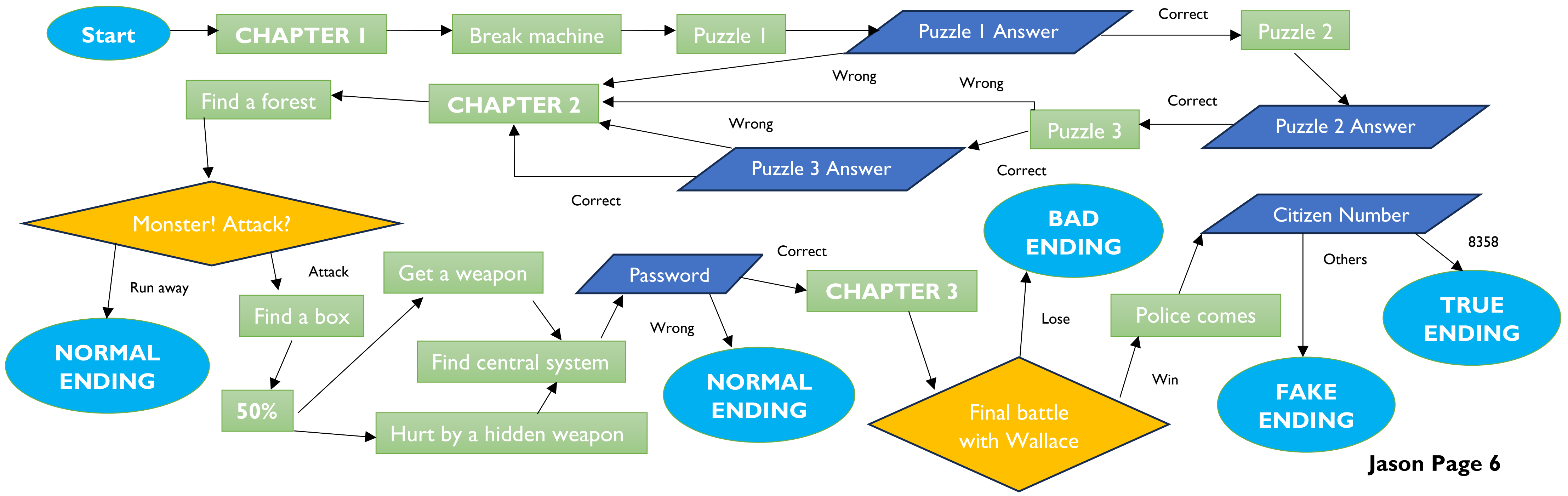
## 3. Flowchart

***Cub's story line:***





***__Wallace's story line:__***



There are 4 types of ending in this game: Bad ending, Normal ending, fake ending, and true ending. Players need to attempt reach the true ending in order to understand the entire story of *Hoosntee Crisis*!

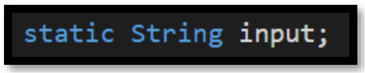
| | **Cub's story line** | **Wallace's story line** |
|---|---|---|
| **BAD ENDING** | Triggered if Health is less than or equal to 0 | |
| **NORMAL ENDING** | Triggered if quit the battle more than 3 times | |
| **FAKE ENDING** | Triggered if user leave partner alone | Triggered if enters other's citizen number |
| **TRUE ENDING** | Triggered if user didn't leave partner | Triggered if enters "8358" |

## 4. Code Explanation & Function

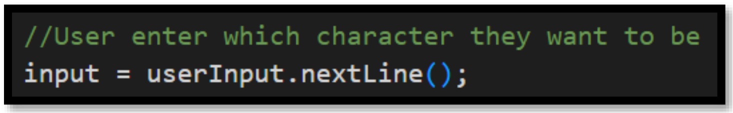
### *4.1 Variables*

#### 4.1.1 Storing user input

For this variable, I have given the identifier called “input”, which also be settled as a static variable. Since the game requires multiple inputs in order to let the game proceed, it is better to let the variable could be access and changed anytime in order for higher efficiency.



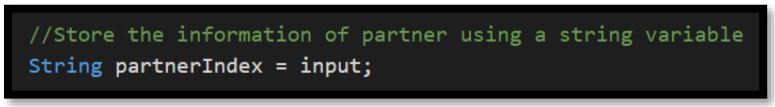
```java
static String input;
```



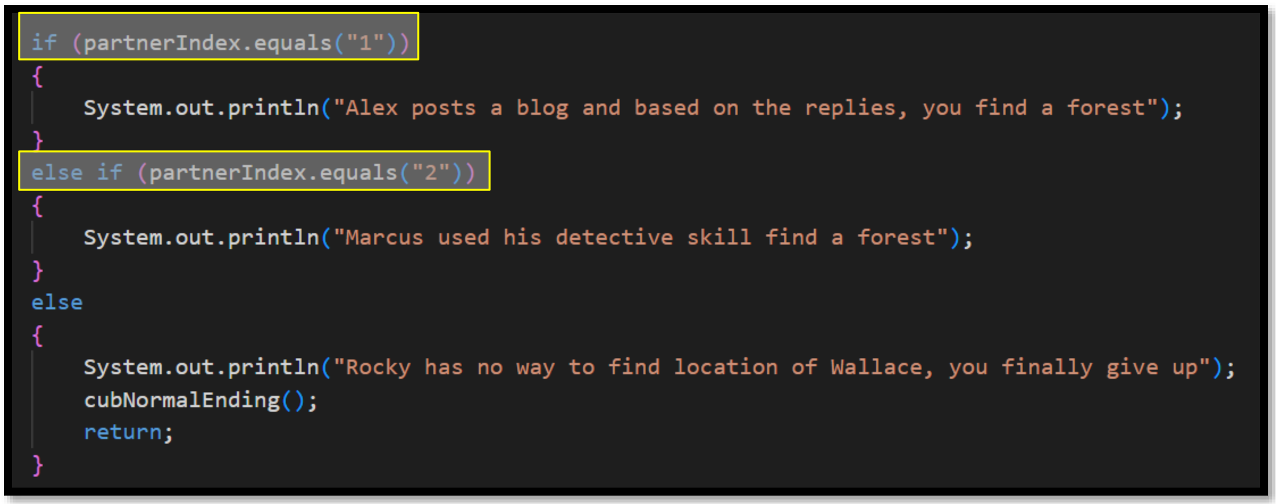
```java
//User enter which character they want to be
input = userInput.nextLine();
```

#### 4.1.2 Storing previous value and utilize in later stage

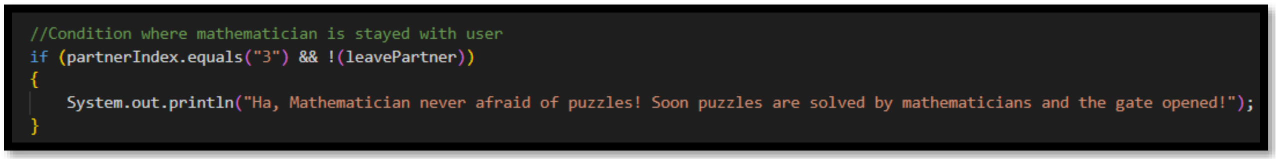
In the program, these kinds of variables are storing for previous entered value by the user, then it will be used to make conditions. For example, `partnerIndex` which was previously given a value, has been used as a condition to create different storyline for choosing partner Alex or choosing partner Marcus. I have named these variables based on the plot.

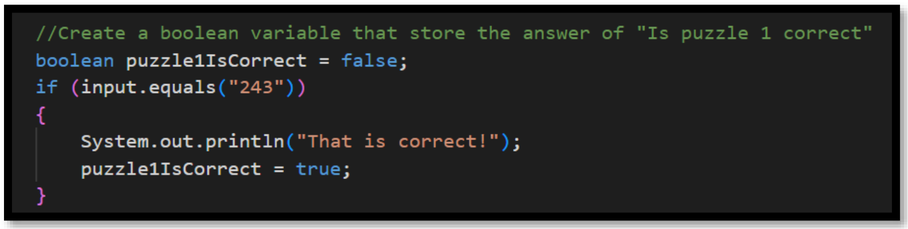


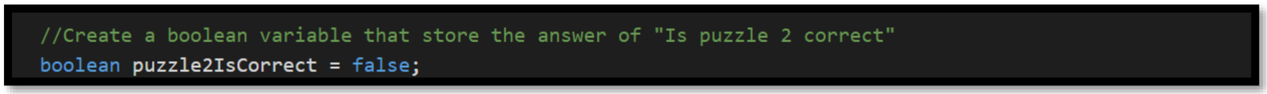
```java
//Store the information of partner using a string variable
String partnerIndex = input;
```

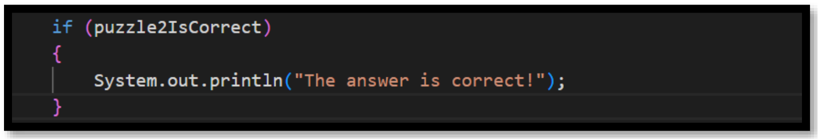


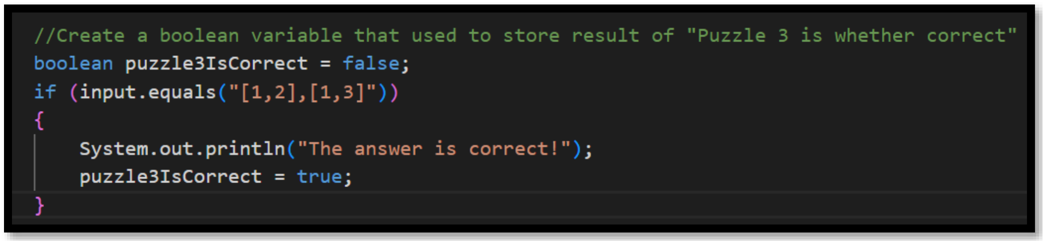
```java
if (partnerIndex.equals("1"))
{
    System.out.println("Alex posts a blog and based on the replies, you find a forest");
}
else if (partnerIndex.equals("2"))
{
    System.out.println("Marcus used his detective skill find a forest");
}
else
{
    System.out.println("Rocky has no way to find location of Wallace, you finally give up");
    cubNormalEnding();
    return;
}
```

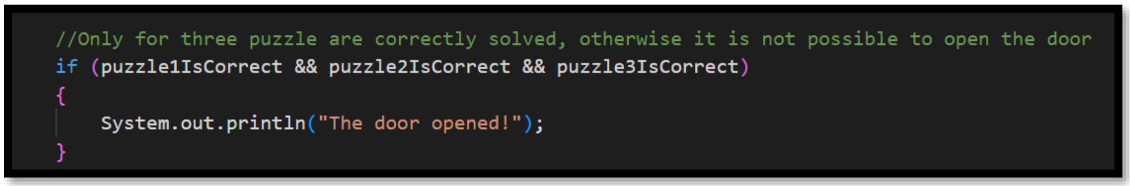

```
//Condition where mathematician is stayed with user
if (partnerIndex.equals("3") && !(leavePartner))
{
    System.out.println("Ha, Mathematician never afraid of puzzles! Soon puzzles are solved by mathematicians and the gate opened!");
}
```


```
//Create a boolean variable that store the answer of "Is puzzle 1 correct"
boolean puzzle1IsCorrect = false;
if (input.equals("243"))
{
    System.out.println("That is correct!");
    puzzle1IsCorrect = true;
}
```


```
//Create a boolean variable that store the answer of "Is puzzle 2 correct"
boolean puzzle2IsCorrect = false;
```

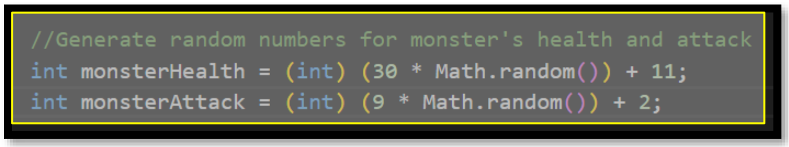

```
if (puzzle2IsCorrect)
{
    System.out.println("The answer is correct!");
}
```


```
//Create a boolean variable that used to store result of "Puzzle 3 is whether correct"
boolean puzzle3IsCorrect = false;
if (input.equals("[1,2],[1,3]"))
{
    System.out.println("The answer is correct!");
    puzzle3IsCorrect = true;
}
```

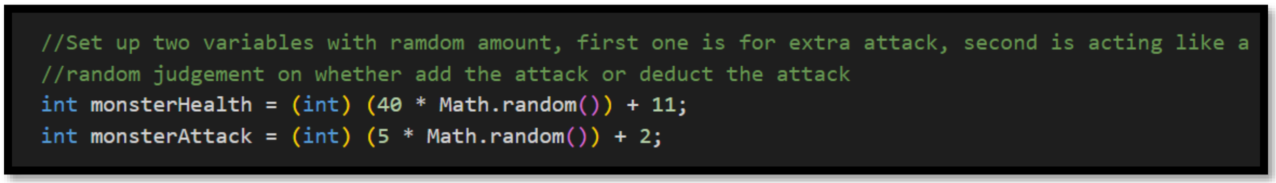

```
//Only for three puzzle are correctly solved, otherwise it is not possible to open the door
if (puzzle1IsCorrect && puzzle2IsCorrect && puzzle3IsCorrect)
{
    System.out.println("The door opened!");
}
```

### 4.1.3 Storing values for complex-structured numbers (Based on codes)

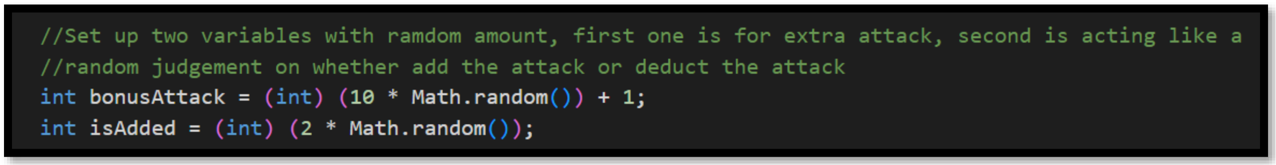
These are the variables which store constant values, but because these constant values are extremely complex, I used a variable in order to make the code much readable. For example, the random numbers for `monsterHealth` and `monsterAttack` have been represented by variables, which makes it clearer to let me know this is monster's health and attack (These 4 codes also used a special method to generate random numbers, which I am going to explain in 2.2).



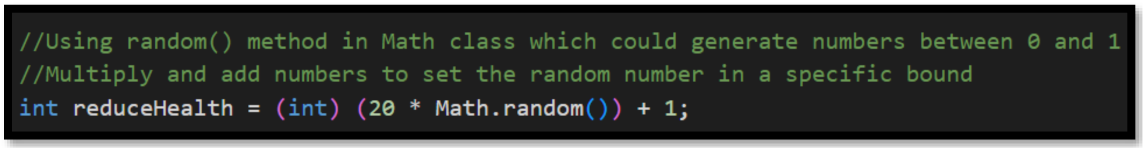
```
//Generate random numbers for monster's health and attack
int monsterHealth = (int) (30 * Math.random()) + 11;
int monsterAttack = (int) (9 * Math.random()) + 2;
```



```
//Set up two variables with ramdom amount, first one is for extra attack, second is acting like a
//random judgement on whether add the attack or deduct the attack
int monsterHealth = (int) (40 * Math.random()) + 11;
int monsterAttack = (int) (5 * Math.random()) + 2;
```



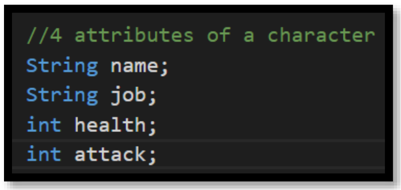
```
//Set up two variables with ramdom amount, first one is for extra attack, second is acting like a
//random judgement on whether add the attack or deduct the attack
int bonusAttack = (int) (10 * Math.random()) + 1;
int isAdded = (int) (2 * Math.random());
```



```
//Using random() method in Math class which could generate numbers between 0 and 1
//Multiply and add numbers to set the random number in a specific bound
int reduceHealth = (int) (20 * Math.random()) + 1;
```

### 4.1.4 Object attributes

There are the variables which store the attributes of characters, which always need to be changed during the battle with monster and boss. I have named these variables based on the attributes every character needs to have. These variables are in the template class, which could be accessed by using object's name + "." + variable's name.

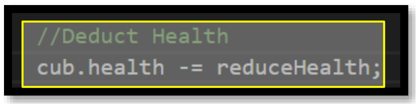


```
//4 attributes of a character
String name;
String job;
int health;
int attack;
```

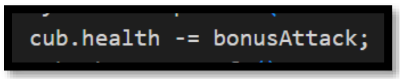
## *4.2 Operators and Random*

### 4.2.1 Operators and operations

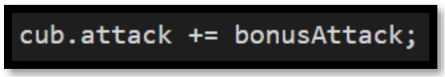
These are some of example that using operators, which are all calculated based on mathematics operations. For example, `cub.health -= reduceHealth`, which is a shorthand version, has been doing operations that makes the variable minus by the value of variable `reduceHealth` to reach the purpose of health deduction.



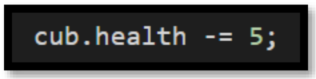
```
//Deduct Health
cub.health -= reduceHealth;
```



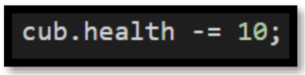
```
cub.health -= bonusAttack;
```



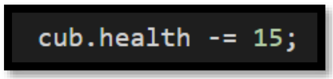
```
cub.attack += bonusAttack;
```



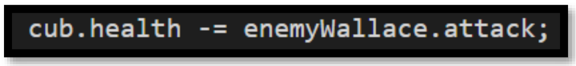
```
cub.health -= 5;
```



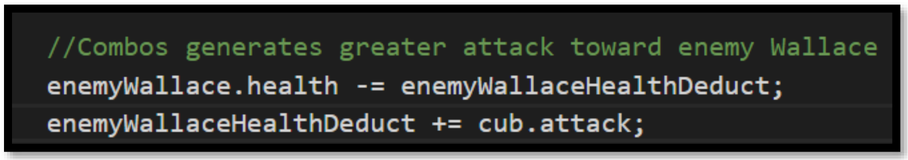
```
cub.health -= 10;
```



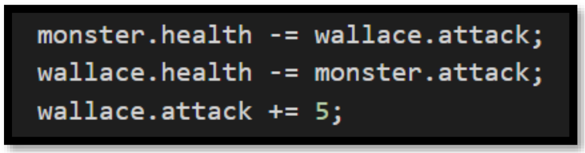
```
cub.health -= 15;
```



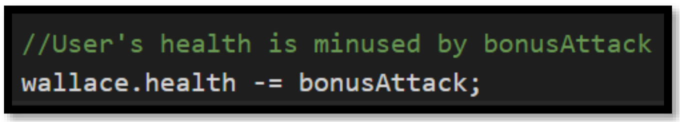
```
cub.health -= enemyWallace.attack;
```



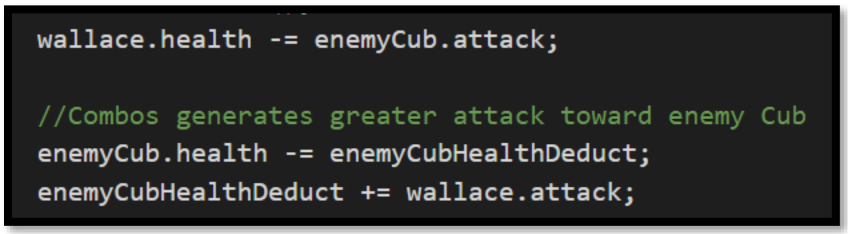
```
//Combos generates greater attack toward enemy Wallace
enemyWallace.health -= enemyWallaceHealthDeduct;
enemyWallaceHealthDeduct += cub.attack;
```



```
monster.health -= wallace.attack;
wallace.health -= monster.attack;
wallace.attack += 5;
```



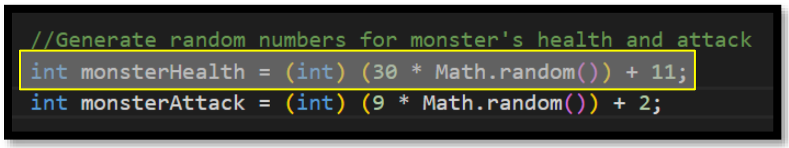
```
//User's health is minused by bonusAttack
wallace.health -= bonusAttack;
```



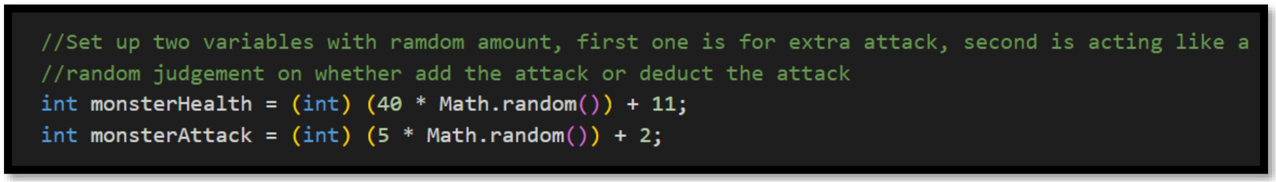
```
wallace.health -= enemyCub.attack;

//Combos generates greater attack toward enemy Cub
enemyCub.health -= enemyCubHealthDeduct;
enemyCubHealthDeduct += wallace.attack;
```

### 4.2.2 Random numbers

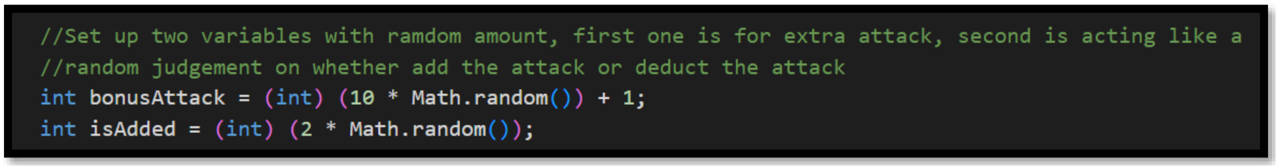
According to research, `Math.random()` is an integer type data which generates a random number between 0 and 1. Through calculation and a cast to integer, any random numbers could be formed. Based on this property, I have introduced this concept into my game, for example, I have given the monster's health with value `(int) (30 * Math.random()) + 11`, which makes all of the value between 11 and 40 possible for monster's health.



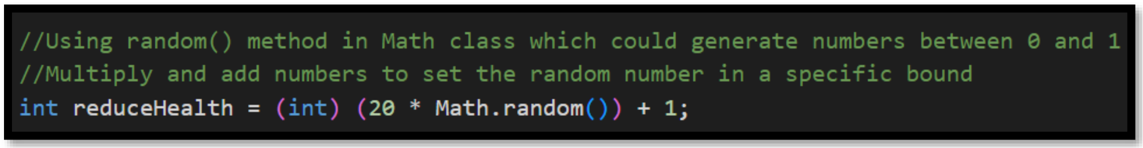
```
//Generate random numbers for monster's health and attack
int monsterHealth = (int) (30 * Math.random()) + 11;
int monsterAttack = (int) (9 * Math.random()) + 2;
```



```
//Set up two variables with ramdom amount, first one is for extra attack, second is acting like a
//random judgement on whether add the attack or deduct the attack
int monsterHealth = (int) (40 * Math.random()) + 11;
int monsterAttack = (int) (5 * Math.random()) + 2;
```



```
//Set up two variables with ramdom amount, first one is for extra attack, second is acting like a
//random judgement on whether add the attack or deduct the attack
int bonusAttack = (int) (10 * Math.random()) + 1;
int isAdded = (int) (2 * Math.random());
```

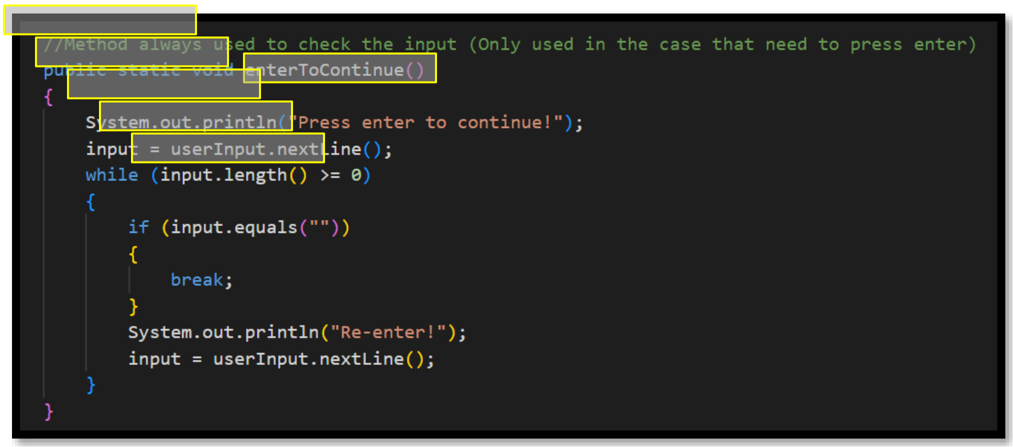


```
//Using random() method in Math class which could generate numbers between 0 and 1
//Multiply and add numbers to set the random number in a specific bound
int reduceHealth = (int) (20 * Math.random()) + 1;
```

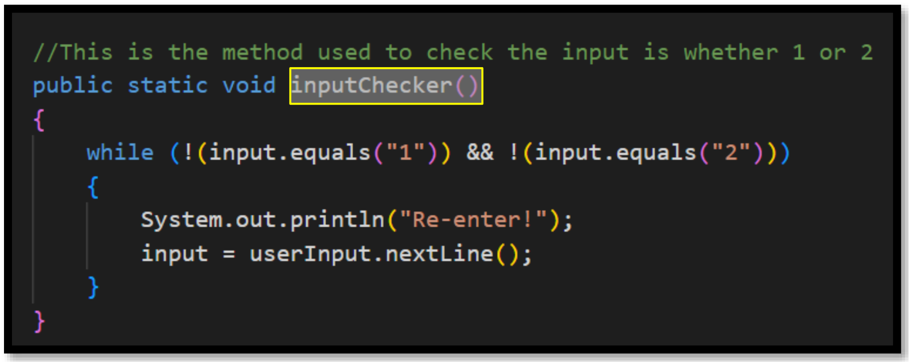
## *4.3 Method*

### 4.3.1 Repeated Codes

These methods possess repeated code throughout the program, which mainly helps the efficiency of the entire code. For example, press enter to continue appears a dozen times, it will be clearer and more efficient if we create a method `enterToContinue()`.



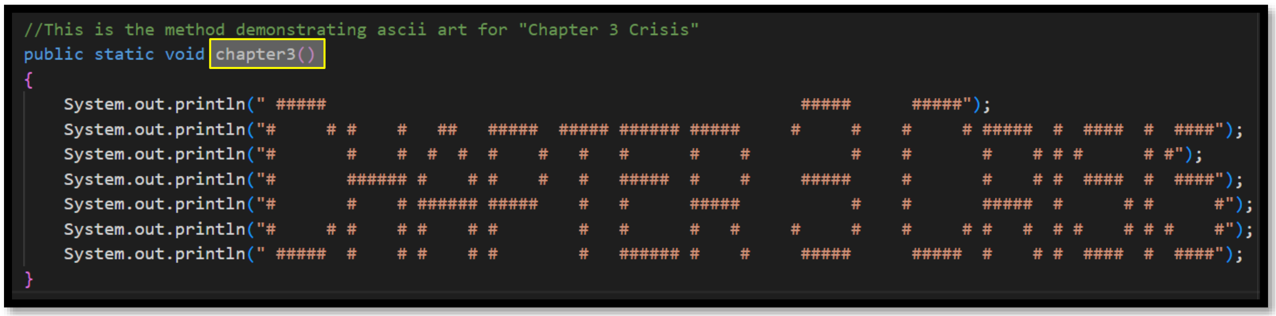
```java
//Method always used to check the input (Only used in the case that need to press enter)
public static void enterToContinue()
{
    System.out.println("Press enter to continue!");
    input = userInput.nextLine();
    while (input.length() >= 0)
    {
        if (input.equals(""))
        {
            break;
        }
        System.out.println("Re-enter!");
        input = userInput.nextLine();
    }
}
```



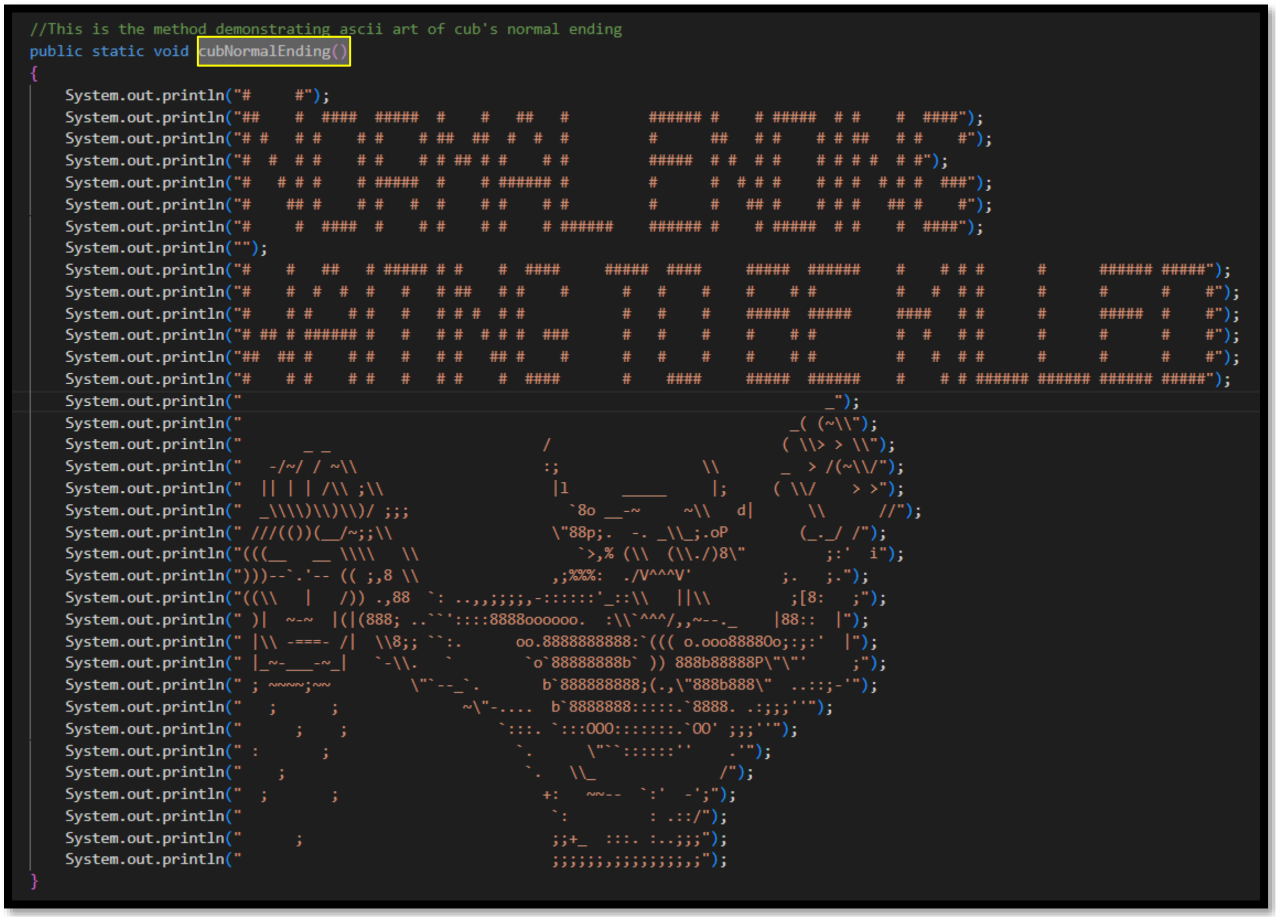
```java
//This is the method used to check the input is whether 1 or 2
public static void inputChecker()
{
    while (!(input.equals("1")) && !(input.equals("2")))
    {
        System.out.println("Re-enter!");
        input = userInput.nextLine();
    }
}
```

### 4.3.2 Repeated Ascii art designs

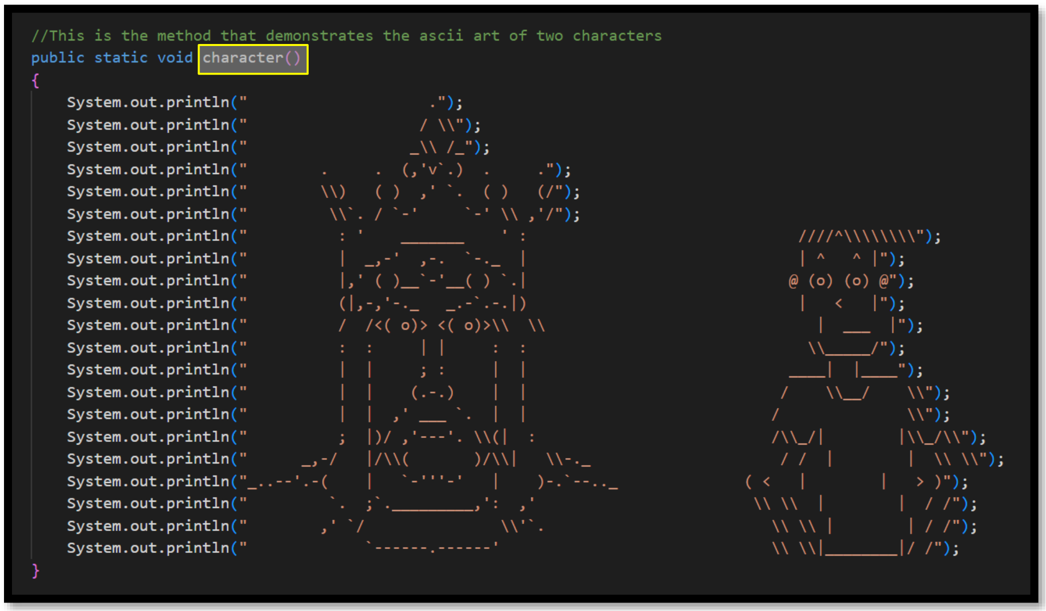
Printing ascii art always requires a lot of lines of code (Like `cubNormalEnding()`, which costs 37 lines), to ensure the efficiency of the program, I have also made some of ascii art be printed using methods.



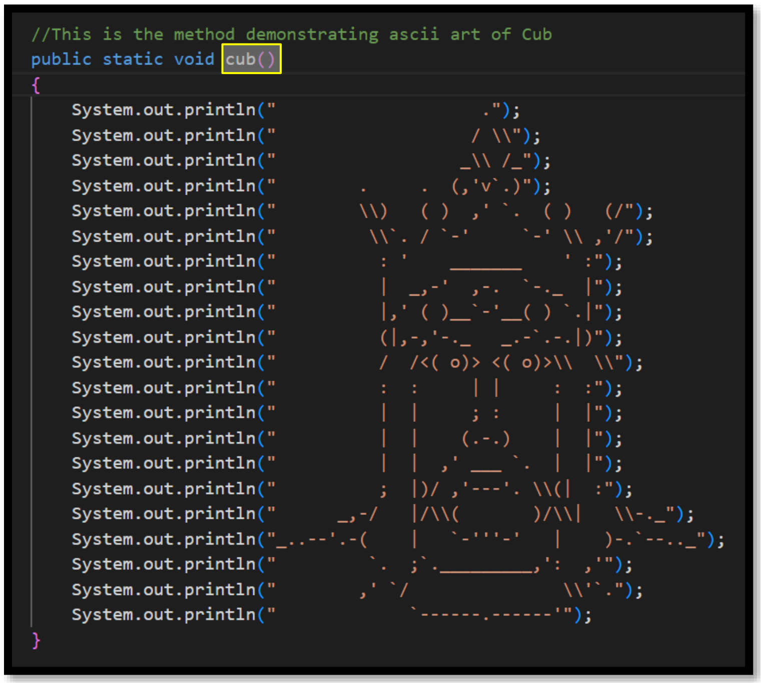
```java
//This is the method demonstrating ascii art for "Chapter 3 Crisis"
public static void chapter3()
{
    System.out.println(" #####                                               #####      #####");
    System.out.println("#     # #    #   ##   #####  ##### ###### #####     #     #    #     # #####  #  ####  #  ####");
    System.out.println("#       #    #  #  #  #    #   #   #      #    #          #    #       #    # # #      # #");
    System.out.println("#       ###### #    # #    #   #   #####  #    #     #####     #       #    # #  ####  #  ####");
    System.out.println("#       #    # ###### #####    #   #      #####          #    #       #####  #      # #      #");
    System.out.println("#     # #    # #    # #        #   #      #   #     #     #    #     # #   #  # #    # # #    #");
    System.out.println(" #####  #    # #    # #        #   ###### #    #     #####      #####  #    # #  ####  #  ####");
}
```



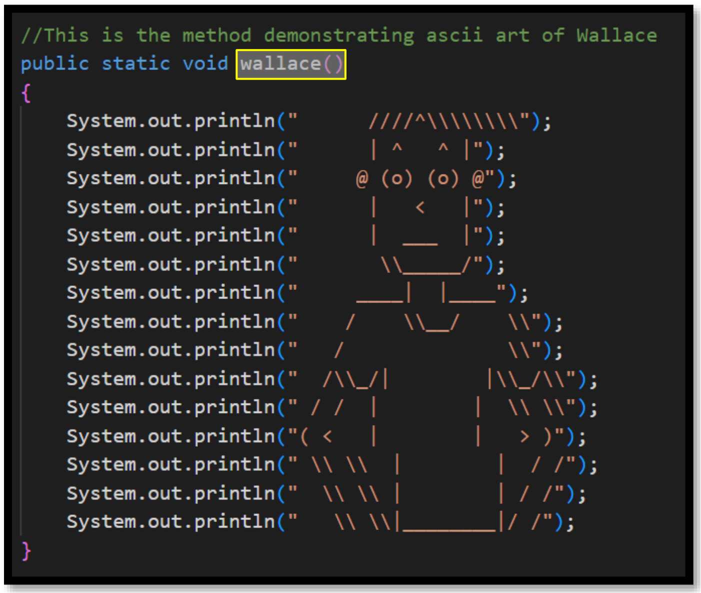
```java
//This is the method demonstrating ascii art of cub's normal ending
public static void cubNormalEnding()
{
    System.out.println("#     #");
    System.out.println("##    #  ####  #####  #    #   ##   #         ###### #    # #####  # #    #  ####");
    System.out.println("# #   # #    # #    # ##  ##  #  #  #         #      ##   # #    # # ##   # #    #");
    System.out.println("#  #  # #    # #    # # ## # #    # #         #####  # #  # #    # # # #  # #");
    System.out.println("#   # # #    # #####  #    # ###### #         #      #  # # #    # # #  # # #  ###");
    System.out.println("#    ## #    # #   #  #    # #    # #         #      #   ## #    # # #   ## #    #");
    System.out.println("#     #  ####  #    # #    # #    # ######    ###### #    # #####  # #    #  ####");
    System.out.println("");
    System.out.println("#    #   ##   # ##### # #    #  ####     #####  ####     #####  ######    #    # # #     #      ###### #####");
    System.out.println("#    #  #  #  #  #   #  # ##   # #    #       #   #    #    #    # #         #   #  # #     #      #      #    #");
    System.out.println("#    # #    # #  #   #  # # #  # #            #   #    #    #####  #####     ####   # #     #      #####  #    #");
    System.out.println("# ## # ###### #  #   #  # #  # # #  ###       #   #    #    #    # #         #  #   # #     #      #      #    #");
    System.out.println("##  ## #    # #  #   #  # #   ## #    #       #   #    #    #    # #         #   #  # #     #      #      #    #");
    System.out.println("#    # #    # #  #   #  # #    #  ####        #    ####     #####  ######    #    # # ###### ###### ###### #####");
    System.out.println("                                                                    _");
    System.out.println("                                                               _( (~\\");
    System.out.println("       _ _                           /                         ( \\> > \\");
    System.out.println("   -/~/ / ~\\                        :;                  \\\\        _  > /(~\\/");
    System.out.println("  || | | /\\ ;\\                      |1        _____     |;       ( \\/    > >");
    System.out.println("  _\\\\)\\)\\)/ ;;;                      `8o __-~     ~\\   d|       \\\\      //");
    System.out.println(" ///(())(__/~;;\\                    \"88p;.  -. _\\_;.oP        (_._/ /");
    System.out.println("(((__   __ \\\\   \\\\                     `>,% (\\  (\\./)8\"         ;:'  i");
    System.out.println(")))--`.'-- (( ;,8 \\\\                  ,;%%%:  ./V^^^V'         ;.   ;.");
    System.out.println("((\\   |   /)) .,88  `: ..,,;;;;;,-::::::'_::\\   ||\\         ;[8:   ;");
    System.out.println(" )|  ~-~  |(|(888; ..``'::::8888oooooo.  :\\`^^^/,,~--._    |88::  |");
    System.out.println(" |\\ -===- /|  \\8;; ``:.      oo.8888888888:`((( o.ooo88880o;:;:'  |");
    System.out.println(" |_~-___-~_|   `-\\.   `        `o`88888888b` )) 888b88888P\"\\"'     ;");
    System.out.println(" ; ~~~~;~~         \"`--_`.       b`888888888;(.,\"888b888\"  ..::;-'");
    System.out.println("   ;       ;             ~\"-....  b`8888888:::::.`8888. .:;;;;''");
    System.out.println("      ;    ;                 `:::. `:::OOO:::::::.`OO' ;;;''");
    System.out.println(" :       ;                      `.      \"``::::::''    .'");
    System.out.println("    ;                            `.   \\_              /");
    System.out.println("  ;        ;                      +:   ~~--  `:'  -';");
    System.out.println("                                   `:          : .::/");
    System.out.println("      ;                            ;;+_  :::. :..;;;");
    System.out.println("                                   ;;;;;;,;;;;;;;;;,;");
}
```



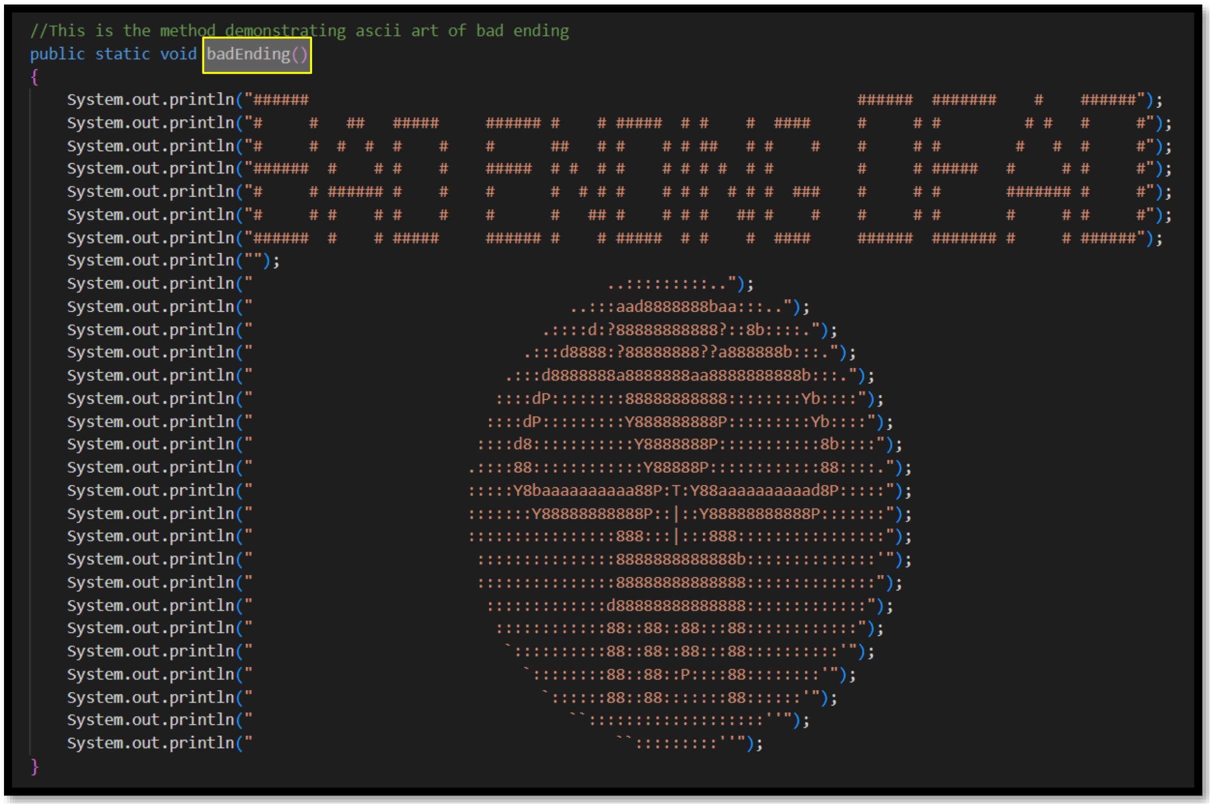
```java
//This is the method that demonstrates the ascii art of two characters
public static void character()
{
    System.out.println("                      .");
    System.out.println("                    / \\");
    System.out.println("                   _\\ /_");
    System.out.println("         .     .  (,'v`.)  .     .");
    System.out.println("        \\)   ( )  ,' `.  ( )   (/");
    System.out.println("         \\`. / `-'     `-' \\ ,'/");
    System.out.println("          : '    _______    ' :                                 ////^\\\\\\\\");
    System.out.println("          |  _,-'  ,-.  `-._  |                                 | ^   ^ |");
    System.out.println("          |,' ( )__`-'__( ) `.|                                @ (o) (o) @");
    System.out.println("          (|,-,'-._   _.-`.-.|)                                 |   <   |");
    System.out.println("          /  /<( o)> <( o)>\\  \\                                  |  ___  |");
    System.out.println("          :  :     | |     :  :                                  \\_____/");
    System.out.println("          |  |     ; :     |  |                                ____|  |____");
    System.out.println("          |  |    (.-.)    |  |                               /    \\__/    \\");
    System.out.println("          |  |  ,' ___ `.  |  |                              /              \\");
    System.out.println("          ;  |)/ ,'---'. \\(|  :                              /\\_/|        |\\_/\\");
    System.out.println("      _,-/   |/\\(        )/\\|   \\-._                         / /  |        |  \\ \\");
    System.out.println("_..--'.-(    |   `-'''-'   |    )-.`--.._                  ( <  |        |  > )");
    System.out.println("         `.  ;`.__________,':  ,'                           \\ \\  |        |  / /");
    System.out.println("        ,' `/              \\'`.                              \\ \\ |        | / /");
    System.out.println("             `------.------'                                 \\ \\|________|/ /");
}
```



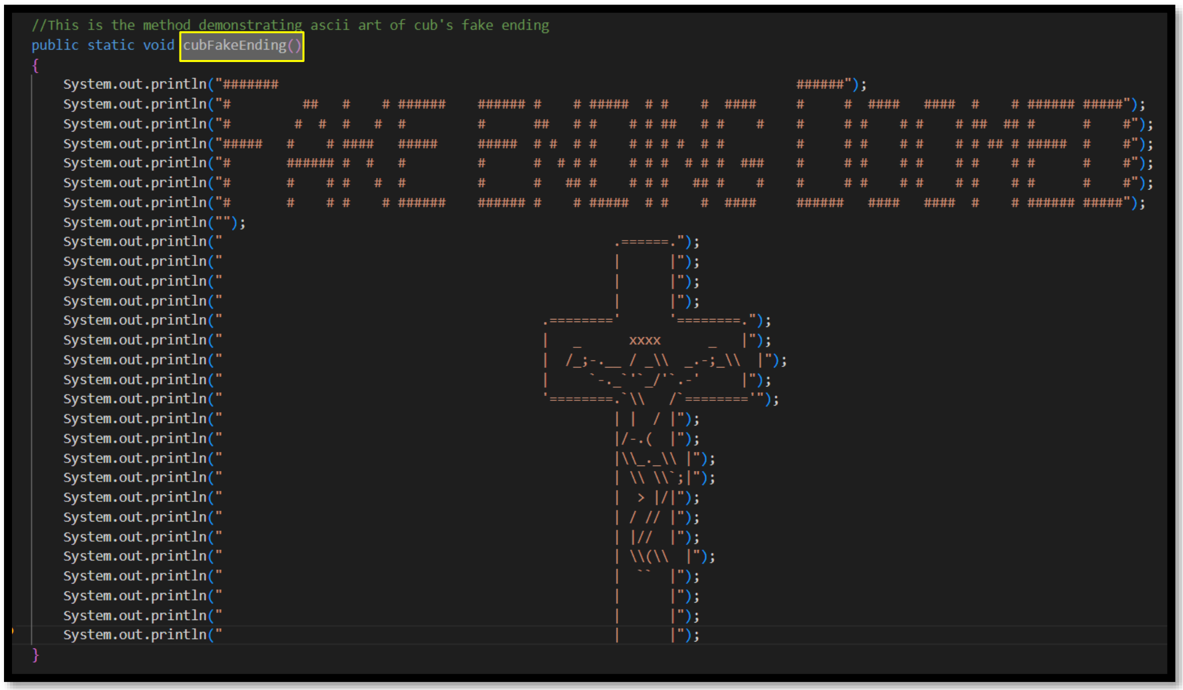
```java
//This is the method demonstrating ascii art of Cub
public static void cub()
{
    System.out.println("                    .");
    System.out.println("                   / \\");
    System.out.println("                  _\\ /_");
    System.out.println("        .     .  (,'v`.)");
    System.out.println("        \\)   ( )  ,' `.  ( )   (/");
    System.out.println("         \\`. / `-'     `-' \\ ,'/");
    System.out.println("          : '    _______    ' :");
    System.out.println("          |  _,-'  ,-.  `-._  |");
    System.out.println("          |,' ( )__`-'__( ) `.|");
    System.out.println("          (|,-,'-._   _.-`.-.|)");
    System.out.println("          /  /<( o)> <( o)>\\  \\");
    System.out.println("          :  :     | |     :  :");
    System.out.println("          |  |     ; :     |  |");
    System.out.println("          |  |    (.-.)    |  |");
    System.out.println("          |  |  ,' ___ `.  |  |");
    System.out.println("          ;  |)/ ,'---'. \\(|  :");
    System.out.println("      _,-/   |/\\(       )/\\|   \\-._");
    System.out.println("_..--'.-(    |   `-'''-'   |    )-.`--.._");
    System.out.println("         `.  ;`._________,':  ,'");
    System.out.println("        ,' `/               \\'`.");
    System.out.println("             `------.------'");
}
```



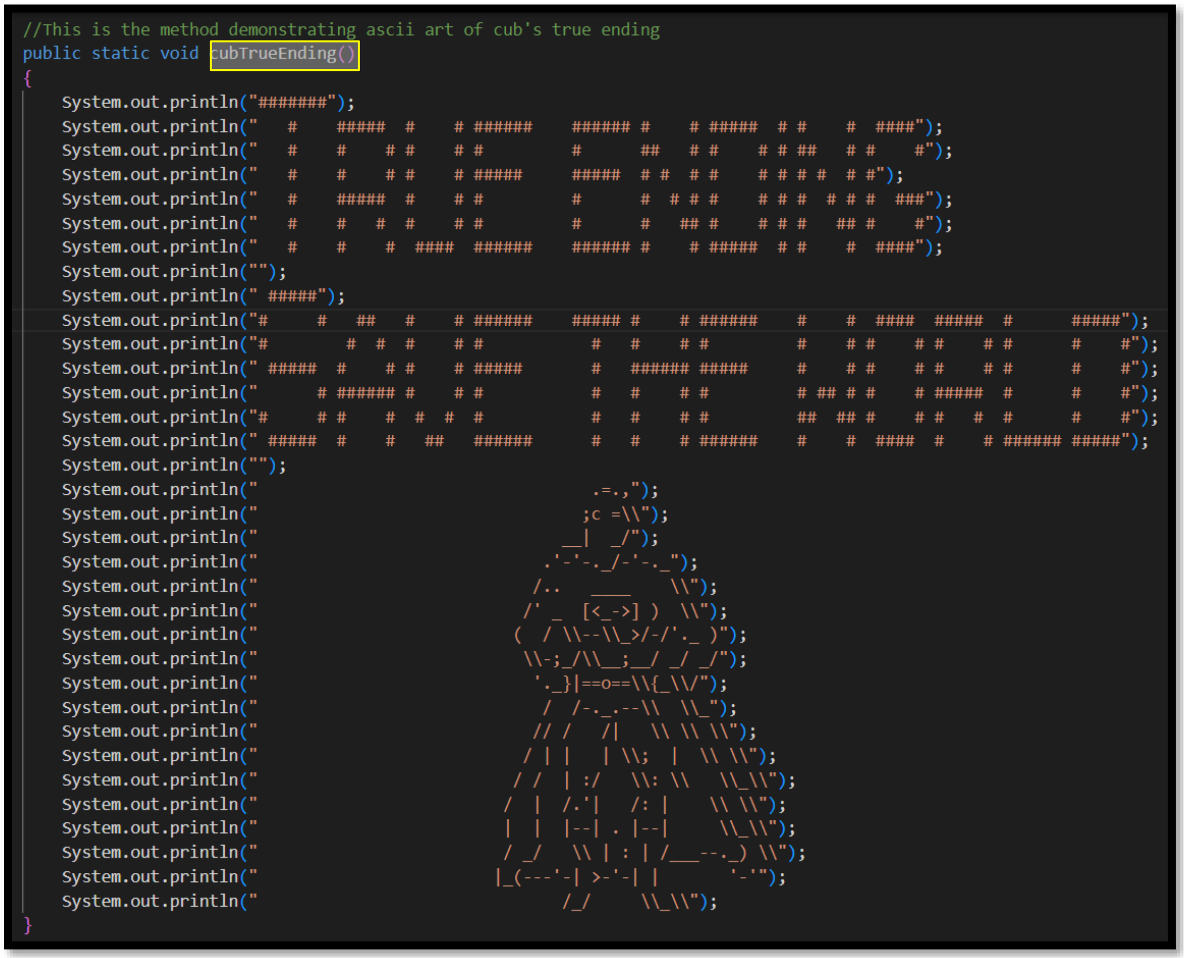
```java
//This is the method demonstrating ascii art of Wallace
public static void wallace()
{
    System.out.println("      ////^\\\\\\\\\\\\\\\\");
    System.out.println("      | ^   ^ |");
    System.out.println("     @ (o) (o) @");
    System.out.println("      |   <   |");
    System.out.println("      |  ___  |");
    System.out.println("       \\_____/");
    System.out.println("     ____|  |____");
    System.out.println("    /    \\__/     \\");
    System.out.println("   /               \\");
    System.out.println("  /\\_/|        |\\_/\\");
    System.out.println(" / /  |        |  \\ \\");
    System.out.println("( <   |        |   > )");
    System.out.println(" \\ \\  |        |  / /");
    System.out.println("  \\ \\ |        | / /");
    System.out.println("   \\ \\|________|/ /");
}
```



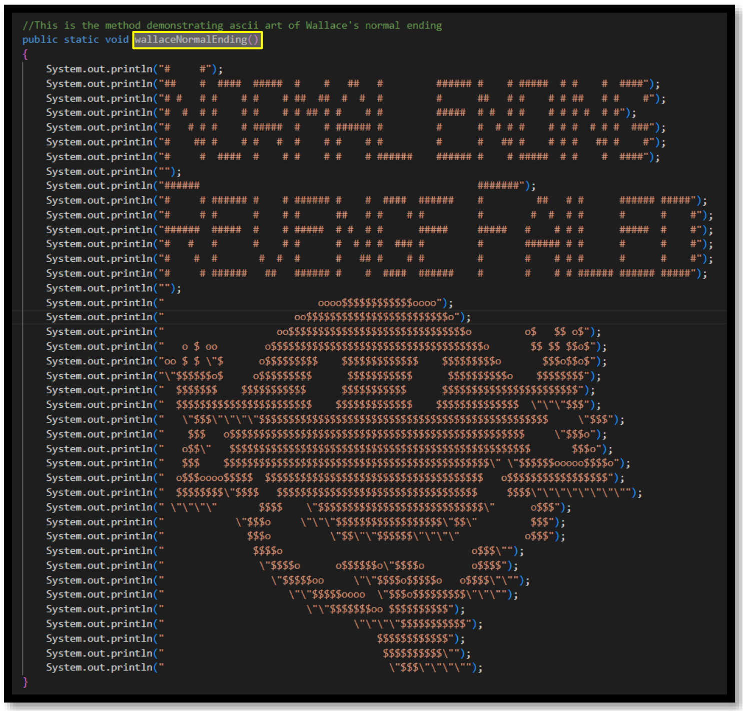
```java
//This is the method demonstrating ascii art of bad ending
public static void badEnding()
{
    System.out.println("######                                                           ######  #######    #    ######");
    System.out.println("#     #   ##   #####     ###### #    # #####  # #    # ####     #     # #         # #   #     #");
    System.out.println("#     #  #  #  #    #    #      ##   # #    # # ##   # #    #    #     # #        #   #  #     #");
    System.out.println("######  #    # #    #    #####  # #  # #    # # # #  # #         #     # #####   #     # #     #");
    System.out.println("#     # ###### #    #    #      #  # # #    # # #  # # #  ###    #     # #       ####### #     #");
    System.out.println("#     # #    # #    #    #      #   ## #    # # #   ## #    #    #     # #       #     # #     #");
    System.out.println("######  #    # #####     ###### #    # #####  # #    #  ####     ######  ####### #     # ######");
    System.out.println("");
    System.out.println("                                      ..:::::::::..");
    System.out.println("                                  ..:::aad8888888baa:::..");
    System.out.println("                               .::::d:?88888888888?::8b::::.");
    System.out.println("                             .:::d8888:?88888888??a888888b:::.");
    System.out.println("                           .:::d8888888a8888888aa8888888888b:::.");
    System.out.println("                          ::::dP::::::::88888888888::::::::Yb::::");
    System.out.println("                         ::::dP:::::::::Y888888888P:::::::::Yb::::");
    System.out.println("                        ::::d8:::::::::::Y8888888P:::::::::::8b::::");
    System.out.println("                       .::::88::::::::::::Y88888P::::::::::::88::::.");
    System.out.println("                       :::::Y8baaaaaaaaaa88P:T:Y88aaaaaaaaaad8P:::::");
    System.out.println("                       :::::::Y88888888888P::|::Y88888888888P:::::::");
    System.out.println("                       ::::::::::::::::888:::|:::888::::::::::::::::");
    System.out.println("                        ::::::::::::::::8888888888888b::::::::::::::'");
    System.out.println("                        :::::::::::::::88888888888888::::::::::::::");
    System.out.println("                         :::::::::::::d88888888888888:::::::::::::");
    System.out.println("                          ::::::::::::88::88::88:::88::::::::::::");
    System.out.println("                           `::::::::::88::88::88:::88::::::::::'");
    System.out.println("                             `::::::::88::88::P::::88::::::::'");
    System.out.println("                               `::::::88::88:::::::88::::::'");
    System.out.println("                                  ``:::::::::::::::::::''");
    System.out.println("                                       ``:::::::::''");
}
```



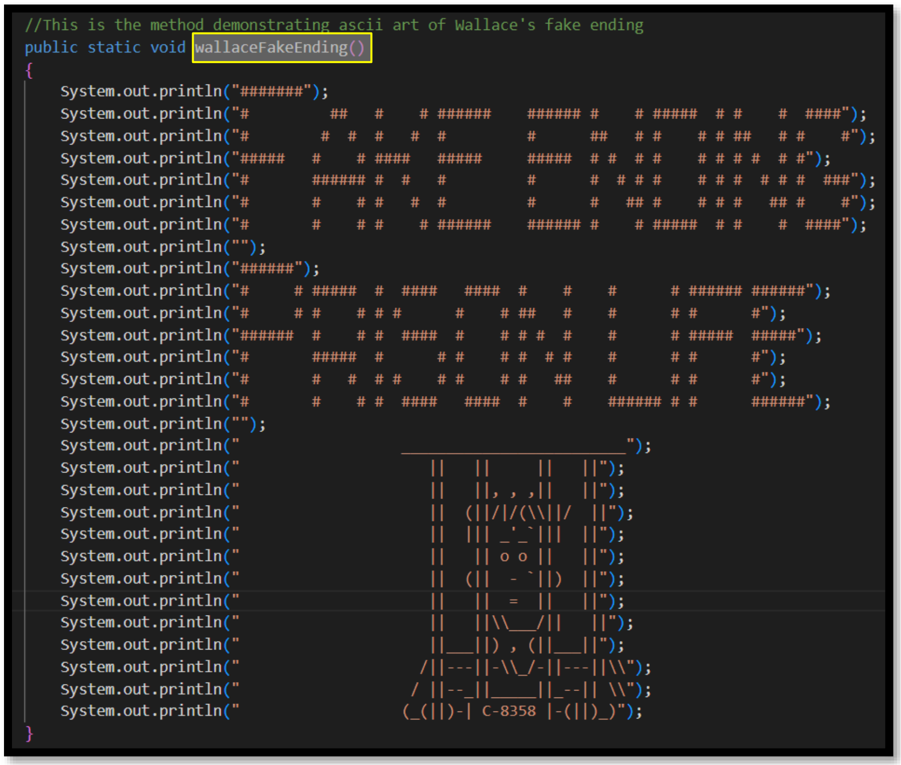
```java
//This is the method demonstrating ascii art of cub's fake ending
public static void cubFakeEnding()
{
    System.out.println("#######                                                                 ######");
    System.out.println("#         ##   #    # ######    ###### #    # #####  # #    #  ####     #     #  ####   ####  #    # ###### #####");
    System.out.println("#        #  #  #   #  #         #      ##   # #    # # ##   # #    #    #     # #    # #    # # ##  ## #      #    #");
    System.out.println("#####   #    # ####   #####     #####  # #  # #    # # # #  # #         #     # #    # #    # # ## # #####  #    #");
    System.out.println("#       ###### #  #   #         #      #  # # #    # # #  # # #  ###    #     # #    # #    # #    # #      #    #");
    System.out.println("#       #    # #   #  #         #      #   ## #    # # #   ## #    #    #     # #    # #    # #    # #      #    #");
    System.out.println("#       #    # #    # ######    ###### #    # #####  # #    #  ####     ######   ####   ####  #    # ###### #####");
    System.out.println("");
    System.out.println("                                                 .======.");
    System.out.println("                                                 |      |");
    System.out.println("                                                 |      |");
    System.out.println("                                                 |      |");
    System.out.println("                                        .========'      '========.");
    System.out.println("                                        |   _      xxxx      _   |");
    System.out.println("                                        |  /_;-.__ / _\\  _.-;_\\  |");
    System.out.println("                                        |     `-._`'`_/'`.-'     |");
    System.out.println("                                        '========.`\\    /`========'");
    System.out.println("                                                 | |  / |");
    System.out.println("                                                 |/-.(  |");
    System.out.println("                                                 |\\_._\\ |");
    System.out.println("                                                 | \\ \\`;|");
    System.out.println("                                                 |  > |/|");
    System.out.println("                                                 | / // |");
    System.out.println("                                                 | |//  |");
    System.out.println("                                                 | \\(\\  |");
    System.out.println("                                                 |  ``  |");
    System.out.println("                                                 |      |");
    System.out.println("                                                 |      |");
    System.out.println("                                                 |      |");
}
```



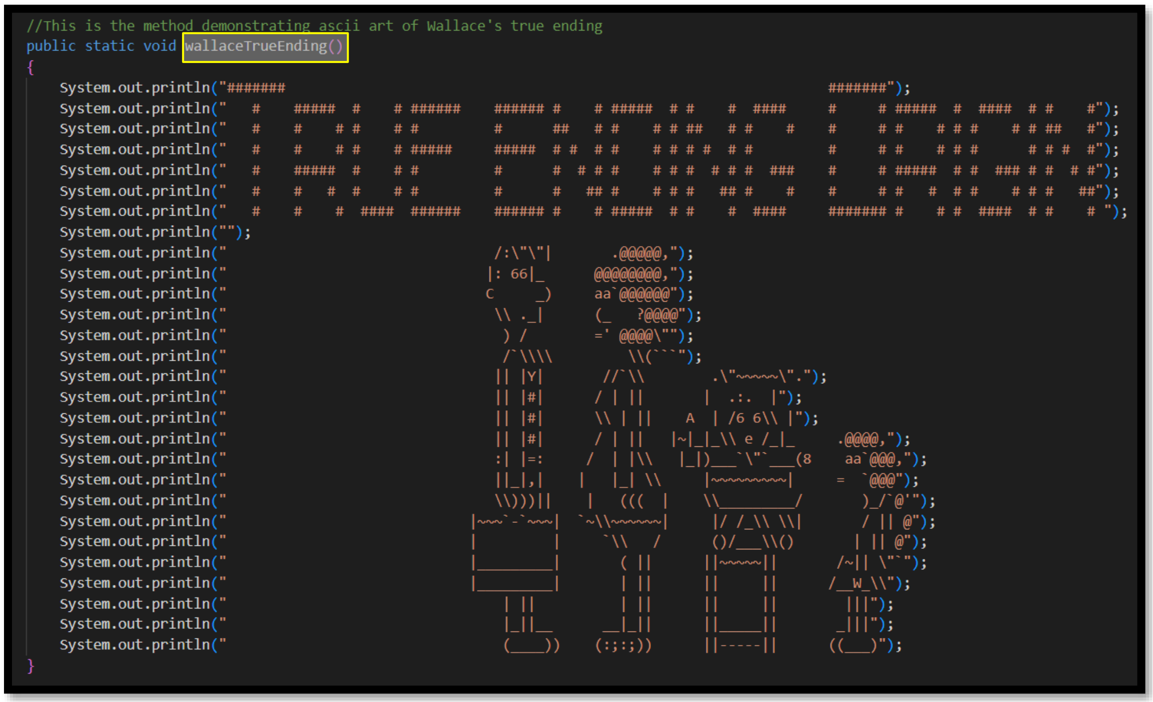
```java
//This is the method demonstrating ascii art of cub's true ending
public static void cubTrueEnding()
{
    System.out.println("#######");
    System.out.println("   #    #####  #    # ######    ###### #    # #####  # #    #  ####");
    System.out.println("   #    #    # #    # #         #      ##   # #    # # ##   # #    #");
    System.out.println("   #    #    # #    # #####     #####  # #  # #    # # # #  # #");
    System.out.println("   #    #####  #    # #         #      #  # # #    # # #  # # #  ###");
    System.out.println("   #    #   #  #    # #         #      #   ## #    # # #   ## #    #");
    System.out.println("   #    #    #  ####  ######    ###### #    # #####  # #    #  ####");
    System.out.println("");
    System.out.println(" #####");
    System.out.println("#     #   ##   #    # ######    ##### #    # ######    #    #  ####  #####   #     #####");
    System.out.println("#        #  #  #    # #           #   #    # #         #    # #    # #    # #      #    #");
    System.out.println(" #####  #    # #    # #####       #   ###### #####     #    # #    # #    # #      #    #");
    System.out.println("      # ###### #    # #           #   #    # #         # ## # #    # #####  #      #    #");
    System.out.println("#     # #    #  #  #  #           #   #    # #         ##  ## #    # #   #  #      #    #");
    System.out.println(" #####  #    #   ##   ######      #   #    # ######    #    #  ####  #    # ###### #####");
    System.out.println("");
    System.out.println("                                  .=.,");
    System.out.println("                                 ;c =\\");
    System.out.println("                               __|  _/");
    System.out.println("                             .'-'-._/-'-._");
    System.out.println("                            /..   ____    \\");
    System.out.println("                           /' _  [<_->] )  \\");
    System.out.println("                          (  / \\--\\_>/-/'._ )");
    System.out.println("                           \\-;_/\\__;__/ _/ _/");
    System.out.println("                            '._}|==o==\\{_\\/");
    System.out.println("                             /  /-._.--\\  \\_");
    System.out.println("                            // /   /|   \\ \\ \\");
    System.out.println("                           / | |   | \\;  |  \\ \\");
    System.out.println("                          / /  | :/   \\: \\   \\_\\");
    System.out.println("                         /  |  /.'|   /: |    \\ \\");
    System.out.println("                         |  |  |--| . |--|    \\_\\");
    System.out.println("                         / _/   \\ | : | /___--._) \\");
    System.out.println("                        |_(---'-| >-'-| |       '-'");
    System.out.println("                               /_/     \\_\\");
}
```



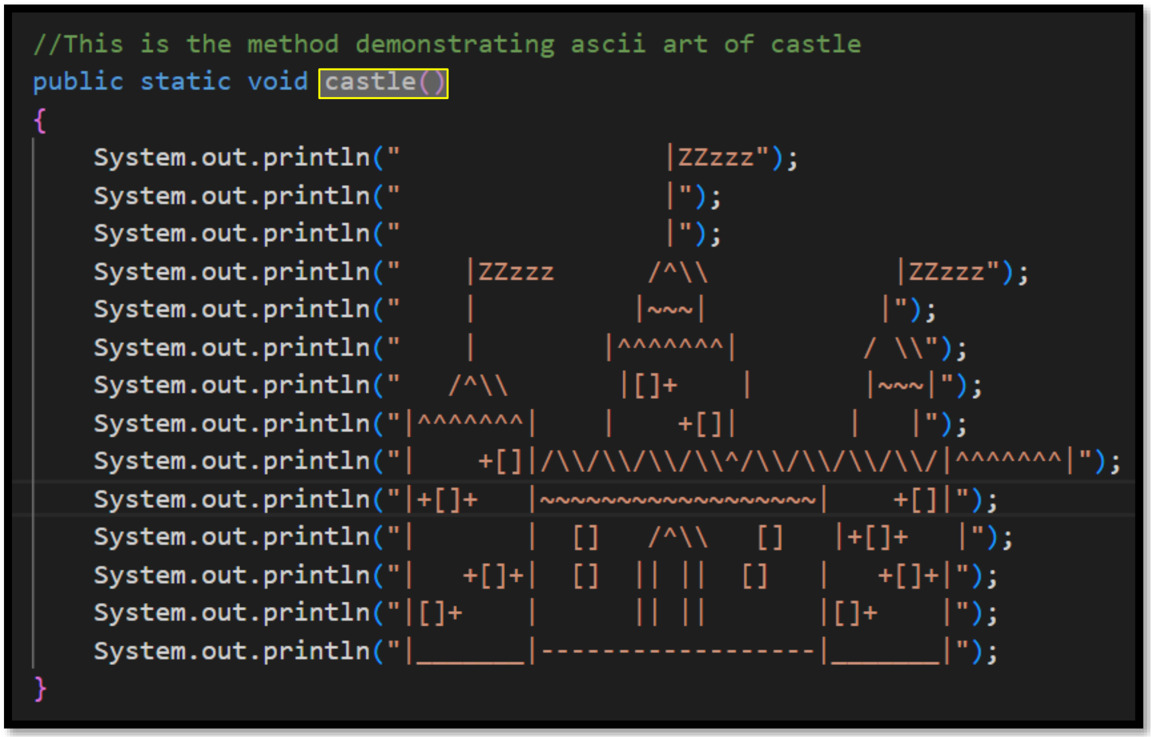
```java
//This is the method demonstrating ascii art of Wallace's normal ending
public static void wallaceNormalEnding()
{
    System.out.println("#      #");
    System.out.println("##     #  ####  #####   #     #   ##    #         ###### #    # #####  # #    #  ####");
    System.out.println("# #    # #    # #     # ##   ##  #  #  #         #      ##   # #    # # ##   # #    #");
    System.out.println("#   #  # #    # #     # #  ## # #    # #         #####  # #  # #    # # # #  # #");
    System.out.println("#    # # #    # #####   #    # ######  #         #      #  # # #    # # #  # # #  ###");
    System.out.println("#    ##  #    # #    #  #     # #    # #         #      #   ## #    # # #   ## #    #");
    System.out.println("#      #  ####  #     # #     # #    # ######    ###### #    # #####  # #    #  ####");
    System.out.println("");
    System.out.println("######                                                   #######");
    System.out.println("#      # ###### #     # ######  #    #  ####  ######     #          ##   # #       ###### #####");
    System.out.println("#      # #      #     # #      ##    # #    # #          #         #  #  # #       #       #    #");
    System.out.println("######   #####  #     # #####  # #   # #      #####      #####    #    # # #       #####   #    #");
    System.out.println("#    #   #      #     # #       #  # # #  ### #          #        ###### # #       #       #    #");
    System.out.println("#     #  #        #  #  #       #   ## #    # #          #        #    # # #       #       #    #");
    System.out.println("#      # ######   ##    ######  #    #  ####  ######     #        #    # # ######  ######  #####");
    System.out.println("");
    System.out.println("                           oooo$$$$$$$$$$$$$oooo");
    System.out.println("                       oo$$$$$$$$$$$$$$$$$$$$$$$$$o");
    System.out.println("                    oo$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$o         o$   $$ o$");
    System.out.println("   o $ oo        o$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$o       $$ $$ $$o$");
    System.out.println("oo $ $ \"$      o$$$$$$$$$    $$$$$$$$$$$$$    $$$$$$$$$o       $$$o$$o$");
    System.out.println("\"$$$$$$o$     o$$$$$$$$$      $$$$$$$$$$$      $$$$$$$$$$o    $$$$$$$$");
    System.out.println("  $$$$$$$    $$$$$$$$$$$      $$$$$$$$$$$      $$$$$$$$$$$$$$$$$$$$$$$");
    System.out.println("  $$$$$$$$$$$$$$$$$$$$$$$    $$$$$$$$$$$$$    $$$$$$$$$$$$$$  \"\"\"$$$");
    System.out.println("   \"$$$\"\"\"\"$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$     \"$$$");
    System.out.println("    $$$   o$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$     \"$$$o");
    System.out.println("   o$$\"   $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$       $$$o");
    System.out.println("   $$$    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\" \"$$$$$$ooooo$$$$o");
    System.out.println("  o$$$oooo$$$$$  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$   o$$$$$$$$$$$$$$$$$");
    System.out.println("  $$$$$$$$\"$$$$   $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$     $$$$\"\"\"\"\"\"\"\"");
    System.out.println(" \"\"\"\"       $$$$    \"$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\"      o$$$");
    System.out.println("            \"$$$o     \"\"\"$$$$$$$$$$$$$$$$$$\"$$\"         $$$");
    System.out.println("              $$$o          \"$$\"\"$$$$$$\"\"\"\"           o$$$");
    System.out.println("               $$$$o                                 o$$$\"");
    System.out.println("                \"$$$$o      o$$$$$$o\"$$$$o        o$$$$");
    System.out.println("                  \"$$$$$oo     \"\"$$$$o$$$$$o   o$$$$\"\"");
    System.out.println("                     \"\"$$$$$oooo  \"$$$o$$$$$$$$$\"\"\"");
    System.out.println("                        \"\"$$$$$$$oo $$$$$$$$$$");
    System.out.println("                                \"\"\"\"$$$$$$$$$$$");
    System.out.println("                                    $$$$$$$$$$$$");
    System.out.println("                                     $$$$$$$$$$\"");
    System.out.println("                                      \"$$$\"\"\"\"");
}
```



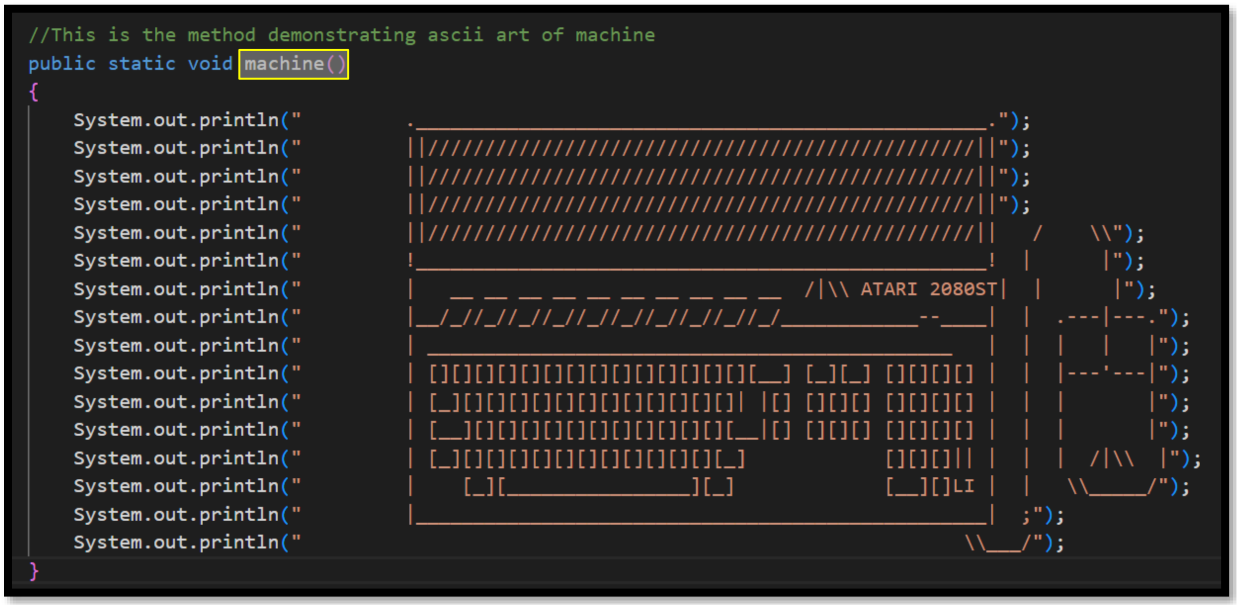
```java
//This is the method demonstrating ascii art of Wallace's fake ending
public static void wallaceFakeEnding()
{
    System.out.println("#######");
    System.out.println("#         ##    #    # ######    ###### #    # #####  # #    #  ####");
    System.out.println("#        #  #  #   #  #         #      ##   # #    # # ##   # #     #");
    System.out.println("#####   #    # ####   #####     #####  # #  # #    # # # #  # #");
    System.out.println("#       ###### #  #   #         #      #  # # #    # # #  # # #  ###");
    System.out.println("#       #    # #   #  #         #      #   ## #    # # #   ## #     #");
    System.out.println("#       #    # #    # ######    ###### #    # #####  # #    #  ####");
    System.out.println("");
    System.out.println("######");
    System.out.println("#     # #####  #  ####   ####  #    #    #      # ###### ######");
    System.out.println("#     # #    # # #     #    # ##   #    #      # #      #");
    System.out.println("######  #    # #  ####  #    # # #  #    #      # #####  #####");
    System.out.println("#       #####  #      # #    # #  # #    #      # #      #");
    System.out.println("#       #   #  # #    # #    # #   ##    #      # #      #");
    System.out.println("#       #    # #  ####   ####  #    #    ###### # #      ######");
    System.out.println("");
    System.out.println("                  __________________________");
    System.out.println("                      ||    ||      ||    ||");
    System.out.println("                      ||    ||, , ,||    ||");
    System.out.println("                      ||   (||/|/(\\||/   ||");
    System.out.println("                      ||   ||| _'_`|||   ||");
    System.out.println("                      ||    || o o ||    ||");
    System.out.println("                      ||   (||  - `||)   ||");
    System.out.println("                      ||    ||  =  ||    ||");
    System.out.println("                      ||    ||\\___/||    ||");
    System.out.println("                      ||___||) , (||___||");
    System.out.println("                     /||---||-\\_/-||---||\\");
    System.out.println("                    / ||--_||_____||_--|| \\");
    System.out.println("                   (_(||)-| C-8358 |-(||)_)");
}
```



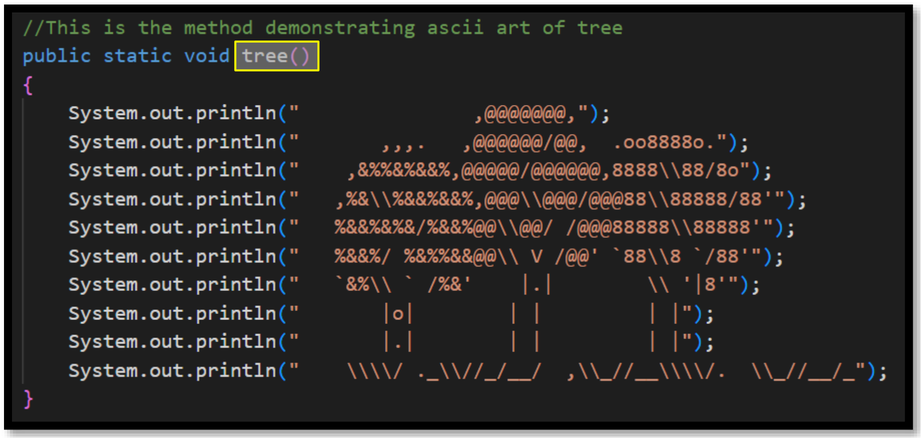
```java
//This is the method demonstrating ascii art of Wallace's true ending
public static void wallaceTrueEnding()
{
    System.out.println("#######                                                              #######");
    System.out.println("   #    #####  #    # ######    ###### #    # #####  # #    #  ####     #     # #####  #  ####  # #    #");
    System.out.println("   #    #    # #    # #         #      ##   # #    # # ##   # #    #     #     # #    # # #     # # ##   #");
    System.out.println("   #    #    # #    # #####     #####  # #  # #    # # # #  # #         #     # #    # #     # # # #   #");
    System.out.println("   #    #####  #    # #         #      #  # # #    # # #  # # #  ###   #     # #####  # ### # # #  # #");
    System.out.println("   #    #   #  #    # #         #      #   ## #    # # #   ## #    #   #     # #   #  # #   # # #   ##");
    System.out.println("   #    #    #  ####  ######    ###### #    # #####  # #    #  ####    ####### #    # #  ####  # #    # ");
    System.out.println("");
    System.out.println("                                 /:\"\"|      .@@@@@,");
    System.out.println("                                |: 66|_     @@@@@@@@,");
    System.out.println("                                C    _)     aa`@@@@@@");
    System.out.println("                                 \\ ._|      (_   ?@@@@");
    System.out.println("                                  ) /       =' @@@@\"");
    System.out.println("                                  /`\\\\          \\(```");
    System.out.println("                                 || |Y|      //`\\          .\"~~~~~\".");
    System.out.println("                                 || |#|     / | ||       |  .:.  |");
    System.out.println("                                 || |#|     \\ | ||   A  | /6 6\\ |");
    System.out.println("                                 || |#|     / | ||  |~|_|_\\ e /_|_     .@@@@,");
    System.out.println("                                 :| |=:    /  | |\\  |_|)___`\"`___(8    aa`@@@,");
    System.out.println("                                 ||_|,|   |   |_| \\     |~~~~~~~~~~|     =  `@@@");
    System.out.println("                                 \\)))||    |   (((  |    \\__________/       )_/`@'");
    System.out.println("                              |~~~`-`~~~|  `~\\~~~~~~~|     |/ /_\\ \\|       / || @");
    System.out.println("                              |         |    `\\   /      ()/___\\()       | || @");
    System.out.println("                              |_________|      ( ||      ||~~~~~||      /~|| \"`");
    System.out.println("                              |_________|      | ||      ||     ||     /__W_\\");
    System.out.println("                                  | ||         | ||      ||     ||       |||");
    System.out.println("                                  |_||__     __|_||      ||_____||      _|||");
    System.out.println("                                  (____))   (:;:;))      ||-----||     ((___)");
}
```



```java
//This is the method demonstrating ascii art of castle
public static void castle()
{
    System.out.println("                 |ZZzzz");
    System.out.println("                 |");
    System.out.println("                 |");
    System.out.println("    |ZZzzz     /^\\            |ZZzzz");
    System.out.println("    |         |~~~|           |");
    System.out.println("    |       |^^^^^^^|        / \\");
    System.out.println("   /^\\       |[]+    |       |~~~|");
    System.out.println("|^^^^^^^|    |    +[]|       |   |");
    System.out.println("|    +[]|/\\/\\/\\/\\^/\\/\\/\\/\\/|^^^^^^^|");
    System.out.println("|+[]+   |~~~~~~~~~~~~~~~~~~|    +[]|");
    System.out.println("|       |  []   /^\\   []   |+[]+   |");
    System.out.println("|   +[]+|  []  || ||  []   |   +[]+|");
    System.out.println("|[]+    |      || ||       |[]+    |");
    System.out.println("|_______|------------------|_______|");
}
```



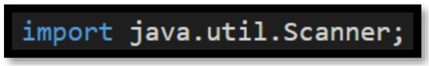
```java
//This is the method demonstrating ascii art of machine
public static void machine()
{
    System.out.println("         .__________________________________________________.");
    System.out.println("         ||////////////////////////////////////////////////||");
    System.out.println("         ||////////////////////////////////////////////////||");
    System.out.println("         ||////////////////////////////////////////////////||");
    System.out.println("         ||////////////////////////////////////////////////||   /    \\");
    System.out.println("         !__________________________________________________!  |      |");
    System.out.println("         |   __ __ __ __ __ __ __ __ __ __  /|\\ ATARI 2080ST|  |      |");
    System.out.println("         |__/_//_//_//_//_//_//_//_//_//_/____________--____|  | .---|---.");
    System.out.println("         | ______________________________________________   |  | |   |   |");
    System.out.println("         | [][][][][][][][][][][][][][][__] [_][_] [][][][] |  | |---'---|");
    System.out.println("         | [_][][][][][][][][][][][][][]| |[] [][][] [][][][] |  | |       |");
    System.out.println("         | [__][][][][][][][][][][][][][__|[] [][][] [][][][] |  | |       |");
    System.out.println("         | [_][][][][][][][][][][][][][_]            [][][]|| |  | |  /|\\  |");
    System.out.println("         |    [_][________________][_]            [__][]LI |  |  \\_____/");
    System.out.println("         |__________________________________________________|  ;");
    System.out.println("                                                          \\___/");
}
```



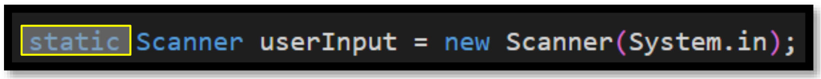
```
//This is the method demonstrating ascii art of tree
public static void tree()
{
    System.out.println("               ,@@@@@@@,");
    System.out.println("       ,,,.   ,@@@@@@/@@,  .oo8888o.");
    System.out.println("    ,&%%&%&&%,@@@@@/@@@@@@,8888\\88/8o");
    System.out.println("   ,%&\\%&&%&&%,@@@\\@@@/@@@88\\88888/88'");
    System.out.println("   %&&%&%&/%&&%@@\\@@/ /@@@88888\\88888'");
    System.out.println("   %&&%/ %&%%&&@@\\ V /@@' `88\\8 `/88'");
    System.out.println("   `&%\\ ` /%&'    |.|        \\ '|8'");
    System.out.println("       |o|        | |         | |");
    System.out.println("       |.|        | |         | |");
    System.out.println("    \\\\/ ._\\//_/__/  ,\\_//__\\\\/.  \\_//__/_");
}
```

## 4.4 Scanner & User input

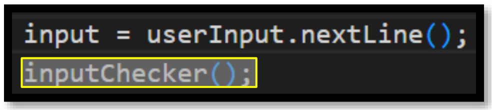
There is only one scanner used in the entire program, which is static and can be accessed any time. This design enhances the efficiency of the program, which prevents creating scanners repeating as program ask for user's input. Input content is also going to be checked by the `inputCheker()`, which is demonstrated in 4.3.1.



```
import java.util.Scanner;
```



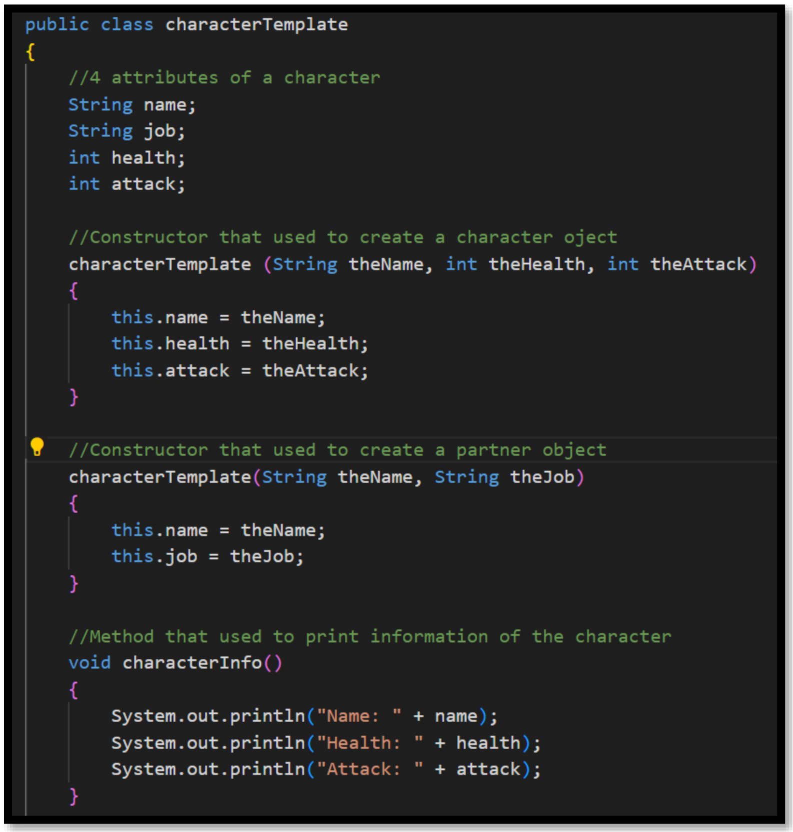
```
static Scanner userInput = new Scanner(System.in);
```



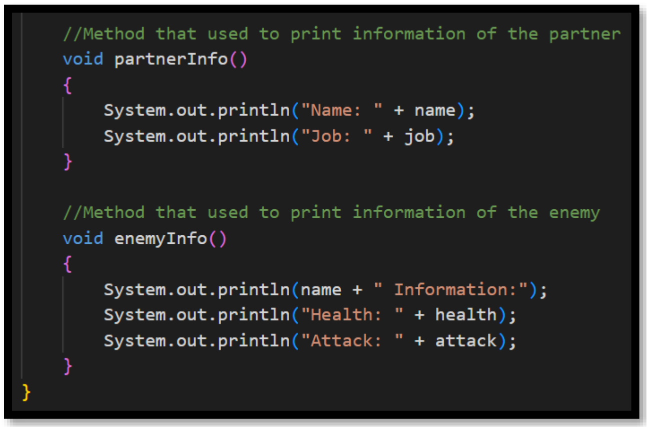
```
input = userInput.nextLine();
inputChecker();
```

## 4.5 Object

<u>*Template:*</u>



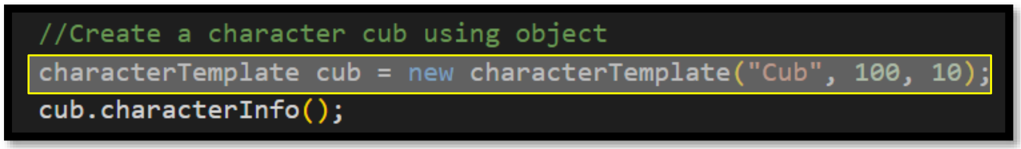
```
public class characterTemplate
{
    //4 attributes of a character
    String name;
    String job;
    int health;
    int attack;

    //Constructor that used to create a character oject
    characterTemplate (String theName, int theHealth, int theAttack)
    {
        this.name = theName;
        this.health = theHealth;
        this.attack = theAttack;
    }

    //Constructor that used to create a partner object
    characterTemplate(String theName, String theJob)
    {
        this.name = theName;
        this.job = theJob;
    }

    //Method that used to print information of the character
    void characterInfo()
    {
        System.out.println("Name: " + name);
        System.out.println("Health: " + health);
        System.out.println("Attack: " + attack);
    }
```



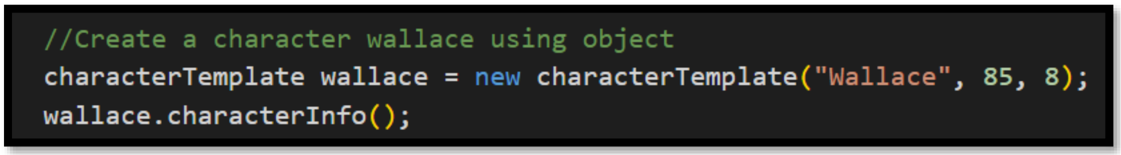
```
    //Method that used to print information of the partner
    void partnerInfo()
    {
        System.out.println("Name: " + name);
        System.out.println("Job: " + job);
    }

    //Method that used to print information of the enemy
    void enemyInfo()
    {
        System.out.println(name + " Information:");
        System.out.println("Health: " + health);
        System.out.println("Attack: " + attack);
    }
}
```

### 4.5.1 Character object

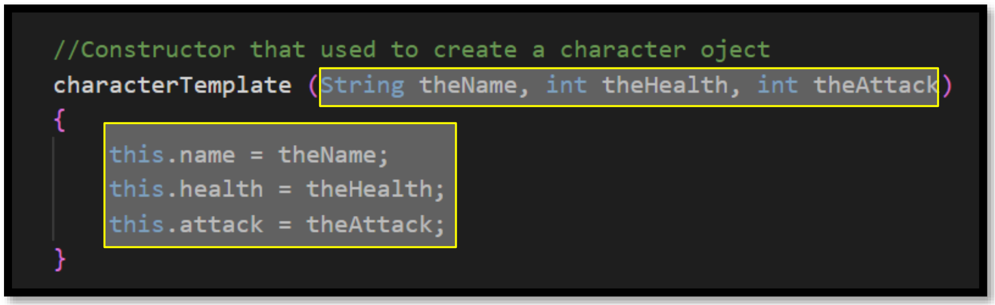
Character objects are mainly used to store and display character information about the protagonist (Cub) and antagonist (Wallace), or in other words, user's health, and attack information. For example, if we are going to create a Cub object, by using the key word `new`, data could directly pass to constructor with formal parameters `theName`, `theHealth` and `theAttack`. As attribute variables in the template have been given value, information about Cub could be directly printed using a designed format for main characters.



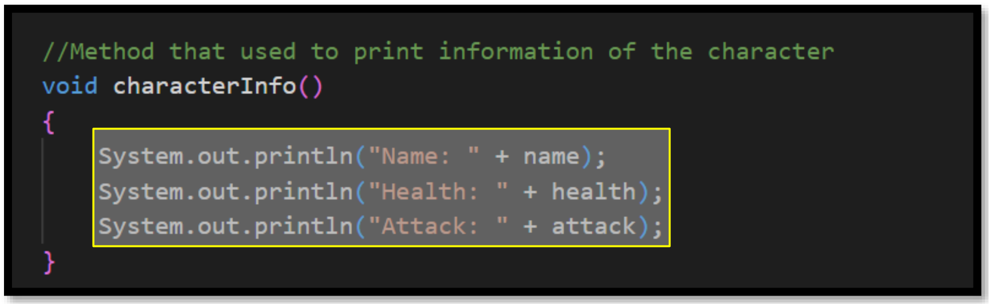
```
//Create a character cub using object
characterTemplate cub = new characterTemplate("Cub", 100, 10);
cub.characterInfo();
```



```
//Create a character wallace using object
characterTemplate wallace = new characterTemplate("Wallace", 85, 8);
wallace.characterInfo();
```



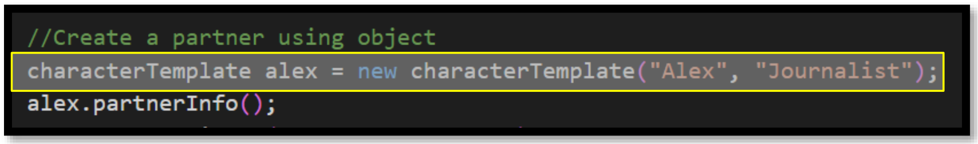
```
//Constructor that used to create a character oject
characterTemplate (String theName, int theHealth, int theAttack)
{
    this.name = theName;
    this.health = theHealth;
    this.attack = theAttack;
}
```



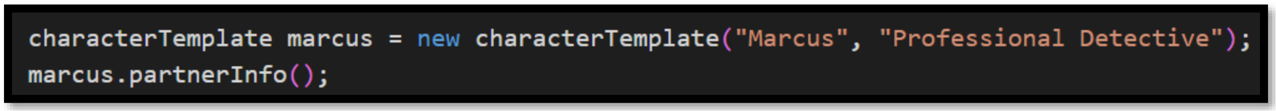
```
//Method that used to print information of the character
void characterInfo()
{
    System.out.println("Name:  " + name);
    System.out.println("Health:  " + health);
    System.out.println("Attack:  " + attack);
}
```

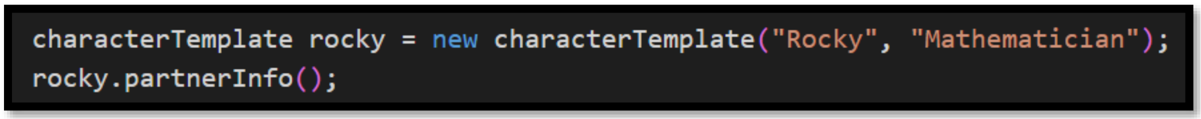
### 4.5.2 Partner Object

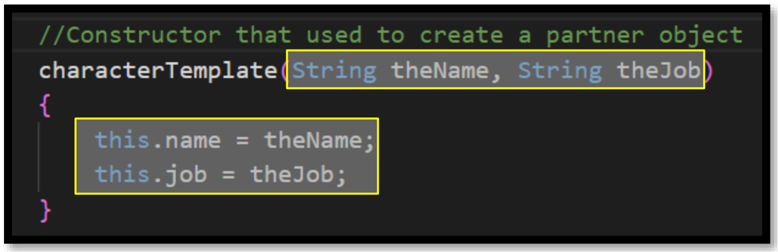
Partner objects only occur in Cub's story line, the object is mainly used to store and display partner information based on method `partnerInfo()`. For example, if user choose Alex be the partner, then the system is going to create a partner object named `alex`, as the same process in 4.5.1, parameters passed to constructor with formal parameters `theName` and `theJob`. As attribute variables in the template have been given value, information about partner could be directly printed using a designed format for partners.

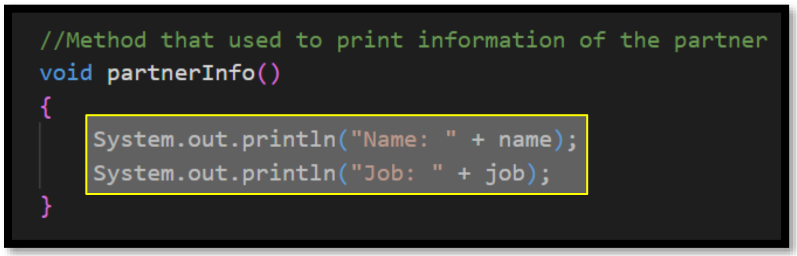


```
//Create a partner using object
characterTemplate alex = new characterTemplate("Alex", "Journalist");
alex.partnerInfo();
```


```
characterTemplate marcus = new characterTemplate("Marcus", "Professional Detective");
marcus.partnerInfo();
```

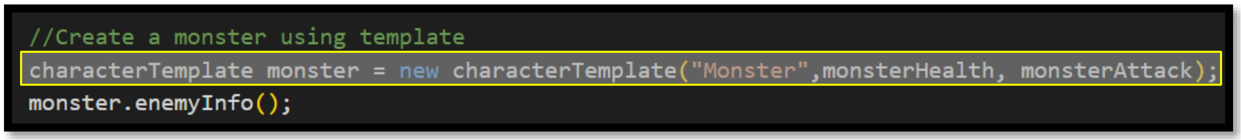

```
characterTemplate rocky = new characterTemplate("Rocky", "Mathematician");
rocky.partnerInfo();
```

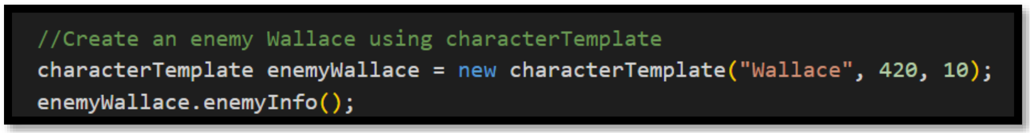

```
//Constructor that used to create a partner object
characterTemplate(String theName, String theJob)
{
    this.name = theName;
    this.job = theJob;
}
```

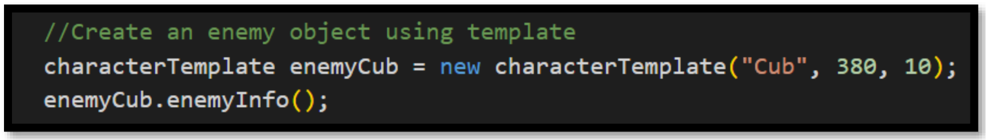

```
//Method that used to print information of the partner
void partnerInfo()
{
    System.out.println("Name:  " + name);
    System.out.println("Job:  " + job);
}
```

### 4.5.3 Enemy Object

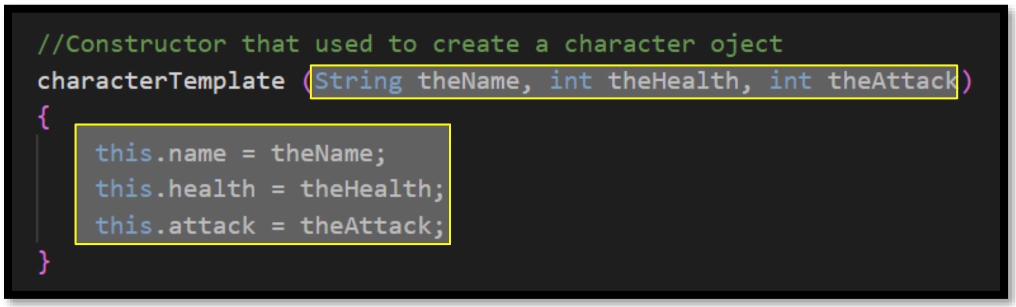
Enemy Objects shares the same constructor with character objects, but the displaying format is different than character objects, that's why I have categorized this separately. For example, if user meet the monster, the system is going to create an enemy object named `monster`, as the same process in 4.5.1, parameters passed to constructor with formal parameters `theName`, `theHealth` and `theAttack`. As attribute variables in the template have been given value, information about monster could be directly printed using a designed format for enemies.


```
//Create a monster using template
characterTemplate monster = new characterTemplate("Monster",monsterHealth, monsterAttack);
monster.enemyInfo();
```

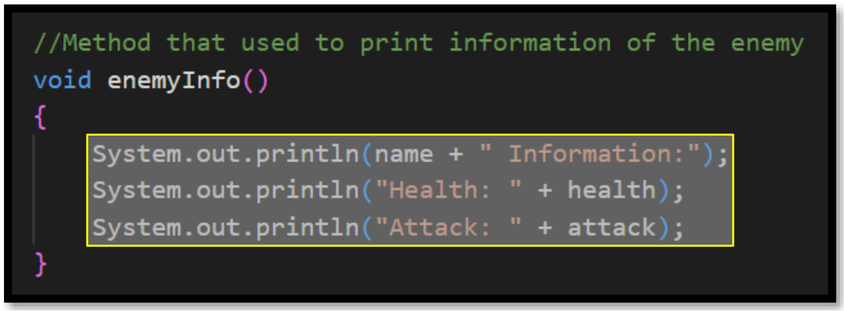

```
//Create an enemy Wallace using characterTemplate
characterTemplate enemyWallace = new characterTemplate("Wallace", 420, 10);
enemyWallace.enemyInfo();
```



```
//Create an enemy object using template
characterTemplate enemyCub = new characterTemplate("Cub", 380, 10);
enemyCub.enemyInfo();
```



```
//Constructor that used to create a character oject
characterTemplate (String theName, int theHealth, int theAttack)
{
    this.name = theName;
    this.health = theHealth;
    this.attack = theAttack;
}
```

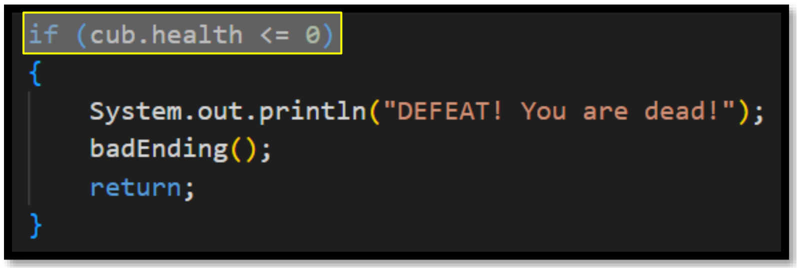


```
//Method that used to print information of the enemy
void enemyInfo()
{
    System.out.println(name + " Information:");
    System.out.println("Health: " + health);
    System.out.println("Attack: " + attack);
}
```

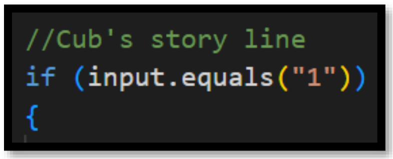
## *4.6. Conditionals*

### 4.6.1 If statements

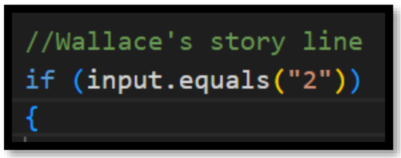
A single if statement is not commonly used in the program, since its main function is to stop the program in specific conditions (reaches the normal ending or bad ending). For example, if statement `if (cub.health <= 0)` has set up the condition that it will leads to the bad ending if variable `cub.health` is less then or equal to 0 during a battle with enemies.



```
if (cub.health <= 0)
{
    System.out.println("DEFEAT! You are dead!");
    badEnding();
    return;
}
```



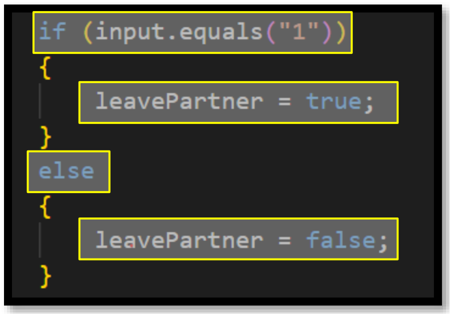
```
//Cub's story line
if (input.equals("1"))
{
```



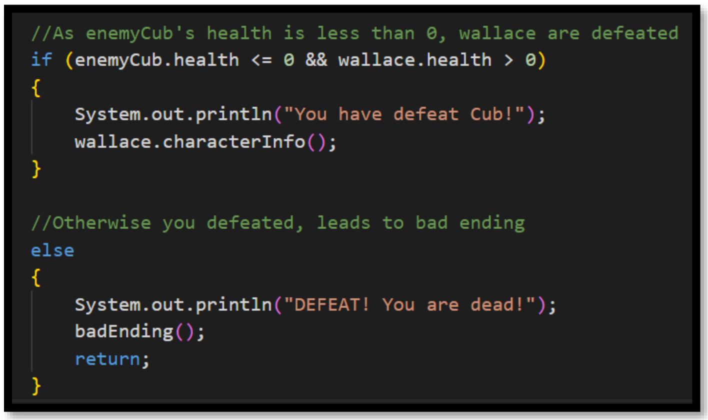
```
//Wallace's story line
if (input.equals("2"))
{
```

### 4.6.2 If else statement

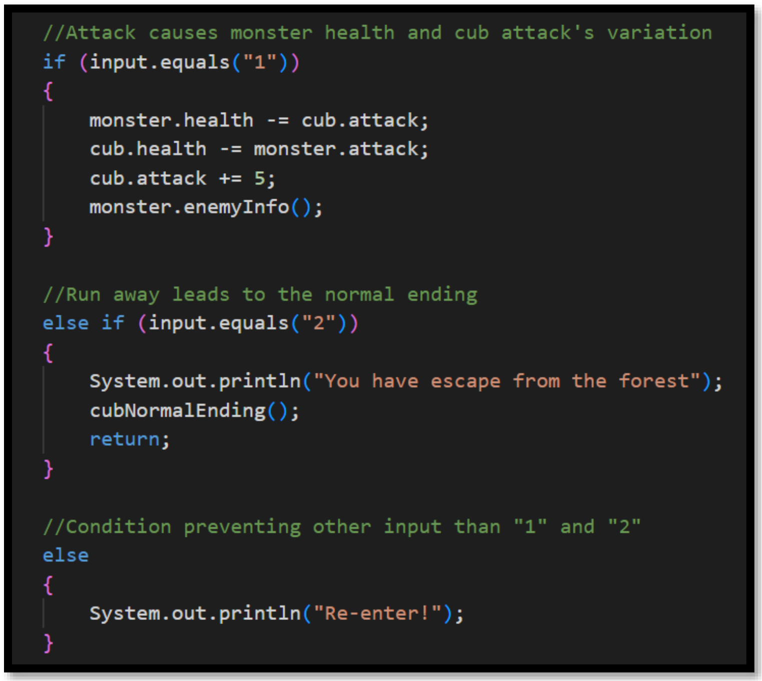
If else statements are the most frequently appeared conditionals throughout the entire program, since such kind of statements makes me could only consider conditions to make program continue without considering the opposite case. This kinds of statements also improves the efficiency of the program, which reduces the number of times on doing conditionals. For example, for if else statement `if (input.equals("1")) else`, when input is equals to 1, it will make the Boolean type variable `leavePartner` to be true, otherwise it be false.

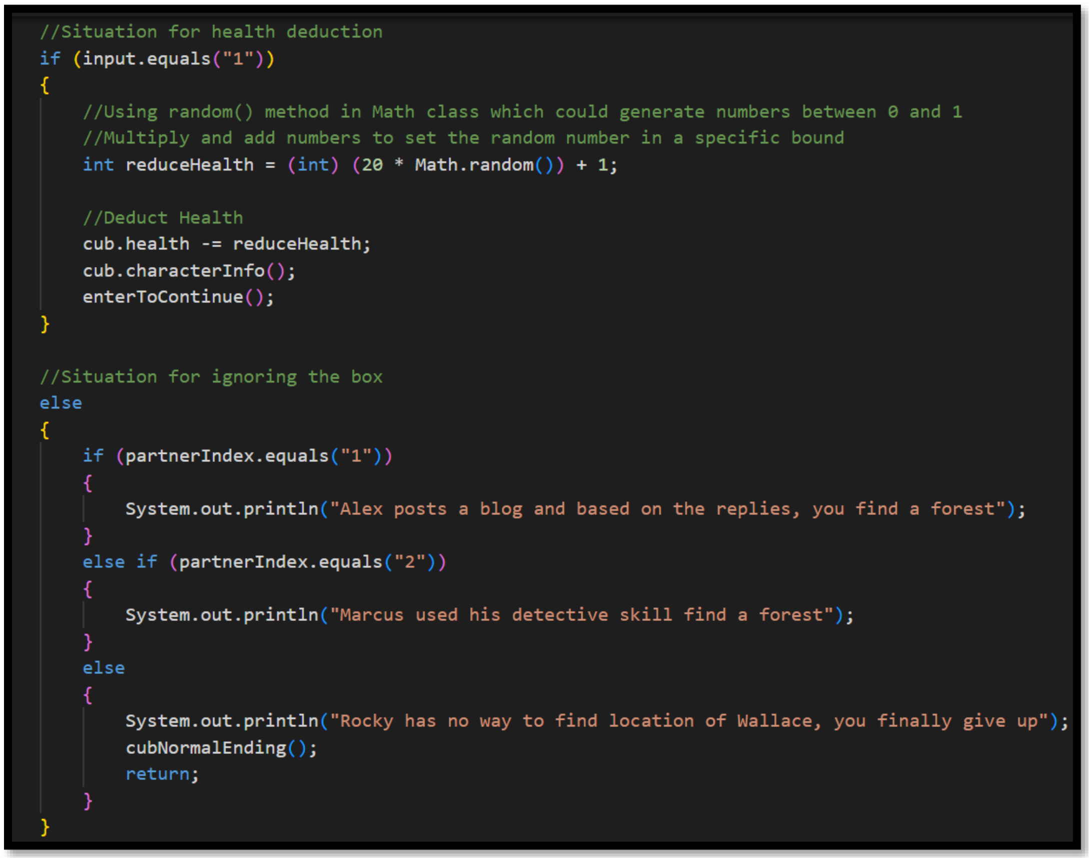


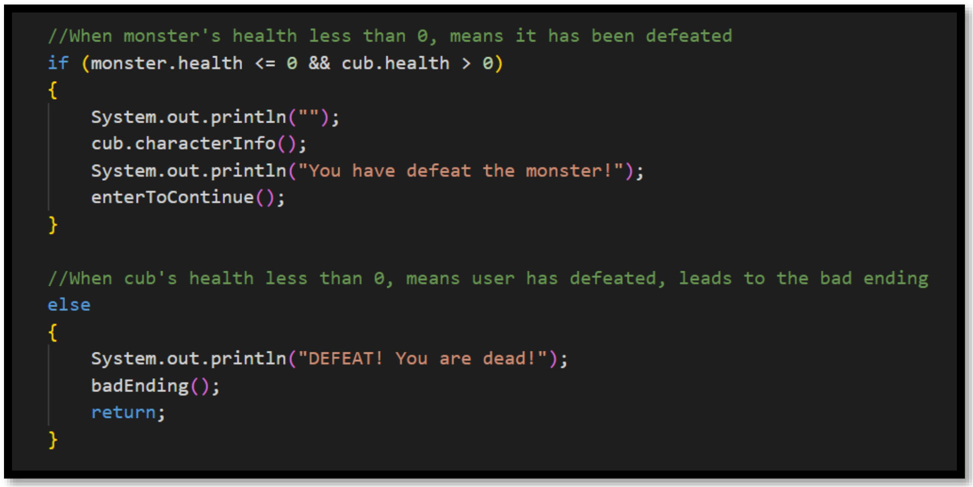
```
if (input.equals("1"))
{
    leavePartner = true;
}
else
{
    leavePartner = false;
}
```



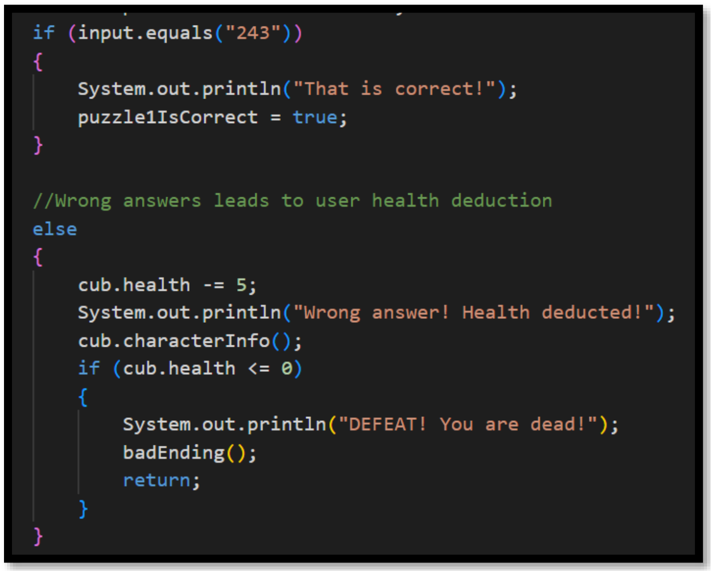
```
//As enemyCub's health is less than 0, wallace are defeated
if (enemyCub.health <= 0 && wallace.health > 0)
{
    System.out.println("You have defeat Cub!");
    wallace.characterInfo();
}

//Otherwise you defeated, leads to bad ending
else
{
    System.out.println("DEFEAT! You are dead!");
    badEnding();
    return;
}
```


```java
//Attack causes monster health and cub attack's variation
if (input.equals("1"))
{
    monster.health -= cub.attack;
    cub.health -= monster.attack;
    cub.attack += 5;
    monster.enemyInfo();
}

//Run away leads to the normal ending
else if (input.equals("2"))
{
    System.out.println("You have escape from the forest");
    cubNormalEnding();
    return;
}

//Condition preventing other input than "1" and "2"
else
{
    System.out.println("Re-enter!");
}
```

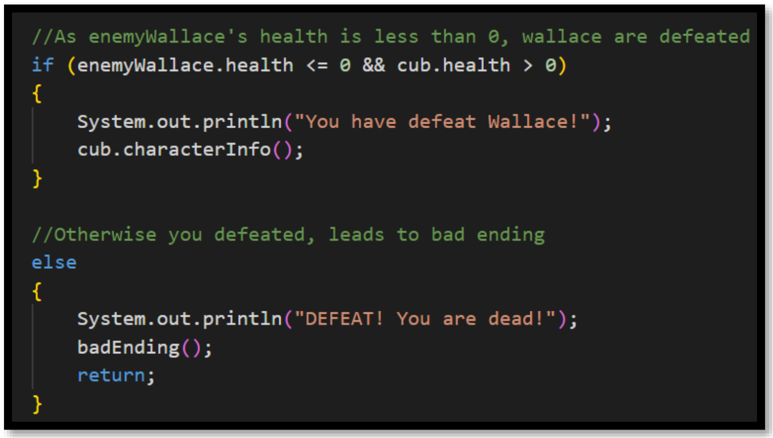

```java
//Situation for health deduction
if (input.equals("1"))
{
    //Using random() method in Math class which could generate numbers between 0 and 1
    //Multiply and add numbers to set the random number in a specific bound
    int reduceHealth = (int) (20 * Math.random()) + 1;

    //Deduct Health
    cub.health -= reduceHealth;
    cub.characterInfo();
    enterToContinue();
}

//Situation for ignoring the box
else
{
    if (partnerIndex.equals("1"))
    {
        System.out.println("Alex posts a blog and based on the replies, you find a forest");
    }
    else if (partnerIndex.equals("2"))
    {
        System.out.println("Marcus used his detective skill find a forest");
    }
    else
    {
        System.out.println("Rocky has no way to find location of Wallace, you finally give up");
        cubNormalEnding();
        return;
    }
}
```



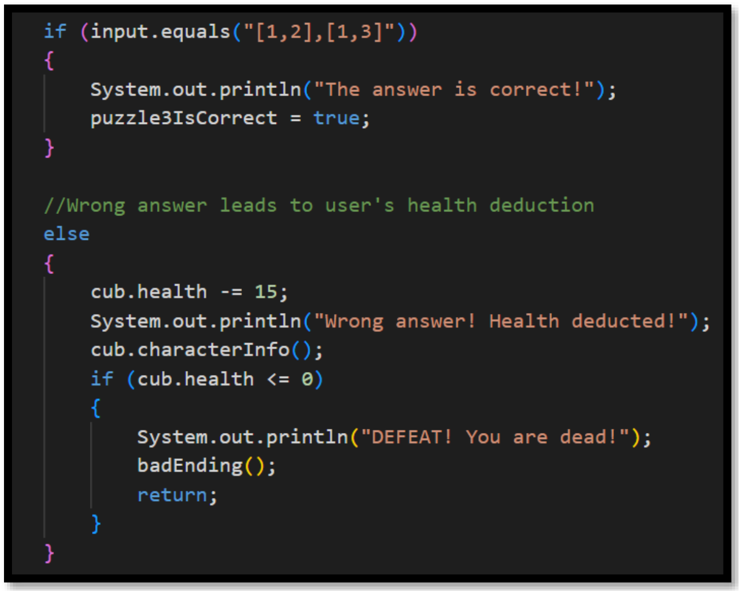
```java
//When monster's health less than 0, means it has been defeated
if (monster.health <= 0 && cub.health > 0)
{
    System.out.println("");
    cub.characterInfo();
    System.out.println("You have defeat the monster!");
    enterToContinue();
}

//When cub's health less than 0, means user has defeated, leads to the bad ending
else
{
    System.out.println("DEFEAT! You are dead!");
    badEnding();
    return;
}
```



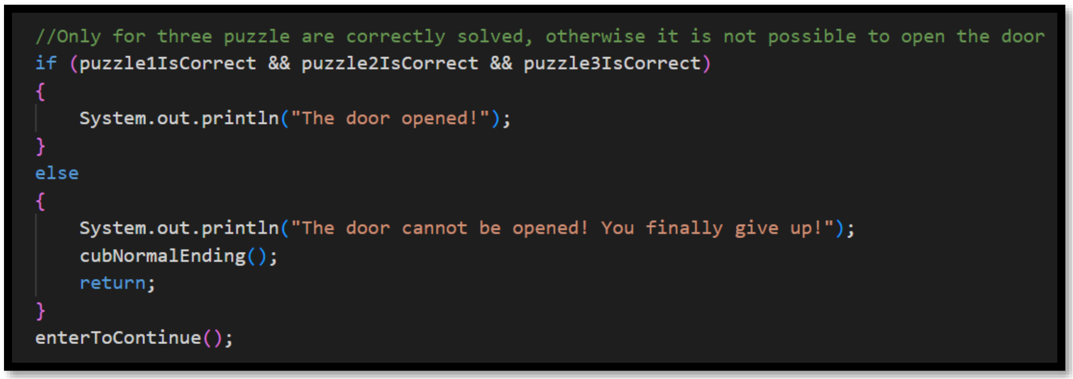
```java
if (input.equals("243"))
{
    System.out.println("That is correct!");
    puzzle1IsCorrect = true;
}

//Wrong answers leads to user health deduction
else
{
    cub.health -= 5;
    System.out.println("Wrong answer! Health deducted!");
    cub.characterInfo();
    if (cub.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        badEnding();
        return;
    }
}
```



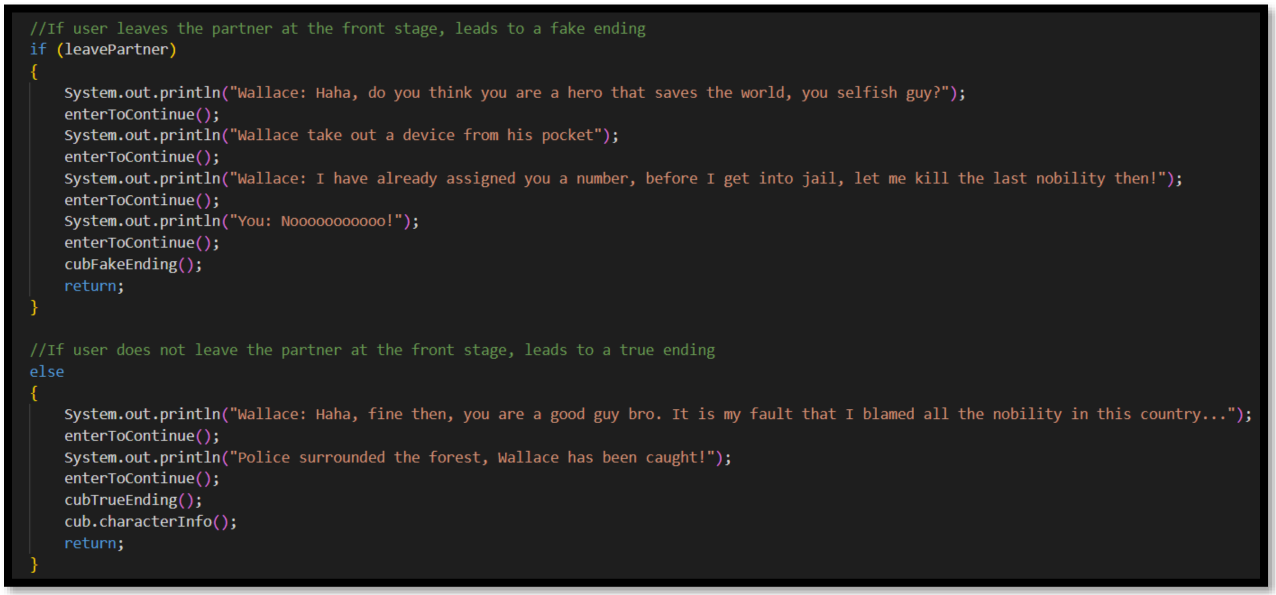
```java
//As enemyWallace's health is less than 0, wallace are defeated
if (enemyWallace.health <= 0 && cub.health > 0)
{
    System.out.println("You have defeat Wallace!");
    cub.characterInfo();
}

//Otherwise you defeated, leads to bad ending
else
{
    System.out.println("DEFEAT! You are dead!");
    badEnding();
    return;
}
```



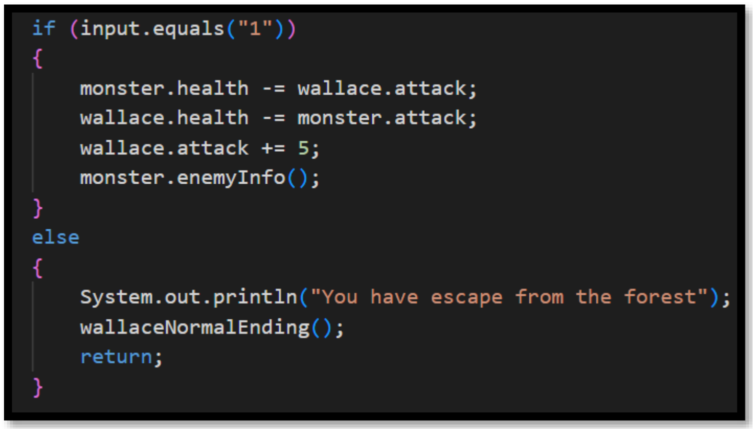
```java
if (input.equals("[1,2],[1,3]"))
{
    System.out.println("The answer is correct!");
    puzzle3IsCorrect = true;
}

//Wrong answer leads to user's health deduction
else
{
    cub.health -= 15;
    System.out.println("Wrong answer! Health deducted!");
    cub.characterInfo();
    if (cub.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        badEnding();
        return;
    }
}
```



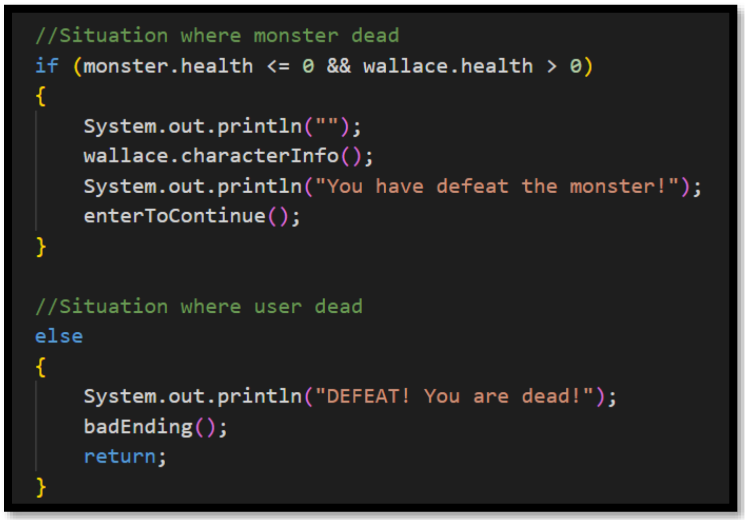
```java
//Only for three puzzle are correctly solved, otherwise it is not possible to open the door
if (puzzle1IsCorrect && puzzle2IsCorrect && puzzle3IsCorrect)
{
    System.out.println("The door opened!");
}
else
{
    System.out.println("The door cannot be opened! You finally give up!");
    cubNormalEnding();
    return;
}
enterToContinue();
```



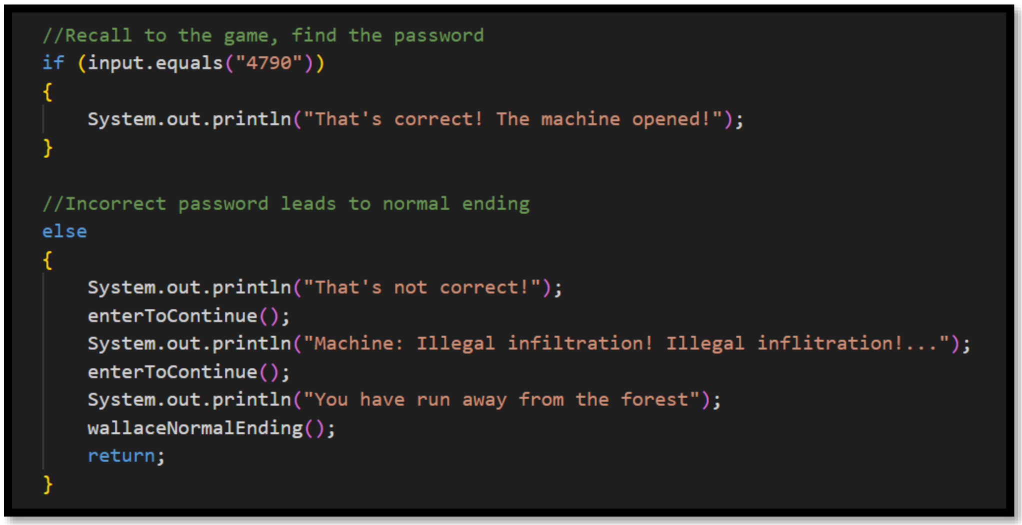
```java
//If user leaves the partner at the front stage, leads to a fake ending
if (leavePartner)
{
    System.out.println("Wallace: Haha, do you think you are a hero that saves the world, you selfish guy?");
    enterToContinue();
    System.out.println("Wallace take out a device from his pocket");
    enterToContinue();
    System.out.println("Wallace: I have already assigned you a number, before I get into jail, let me kill the last nobility then!");
    enterToContinue();
    System.out.println("You: Nooooooooooo!");
    enterToContinue();
    cubFakeEnding();
    return;
}

//If user does not leave the partner at the front stage, leads to a true ending
else
{
    System.out.println("Wallace: Haha, fine then, you are a good guy bro. It is my fault that I blamed all the nobility in this country...");
    enterToContinue();
    System.out.println("Police surrounded the forest, Wallace has been caught!");
    enterToContinue();
    cubTrueEnding();
    cub.characterInfo();
    return;
}
```



```java
if (input.equals("1"))
{
    monster.health -= wallace.attack;
    wallace.health -= monster.attack;
    wallace.attack += 5;
    monster.enemyInfo();
}
else
{
    System.out.println("You have escape from the forest");
    wallaceNormalEnding();
    return;
}
```



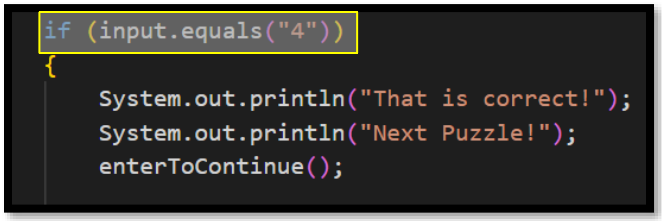
```java
//Situation where monster dead
if (monster.health <= 0 && wallace.health > 0)
{
    System.out.println("");
    wallace.characterInfo();
    System.out.println("You have defeat the monster!");
    enterToContinue();
}

//Situation where user dead
else
{
    System.out.println("DEFEAT! You are dead!");
    badEnding();
    return;
}
```



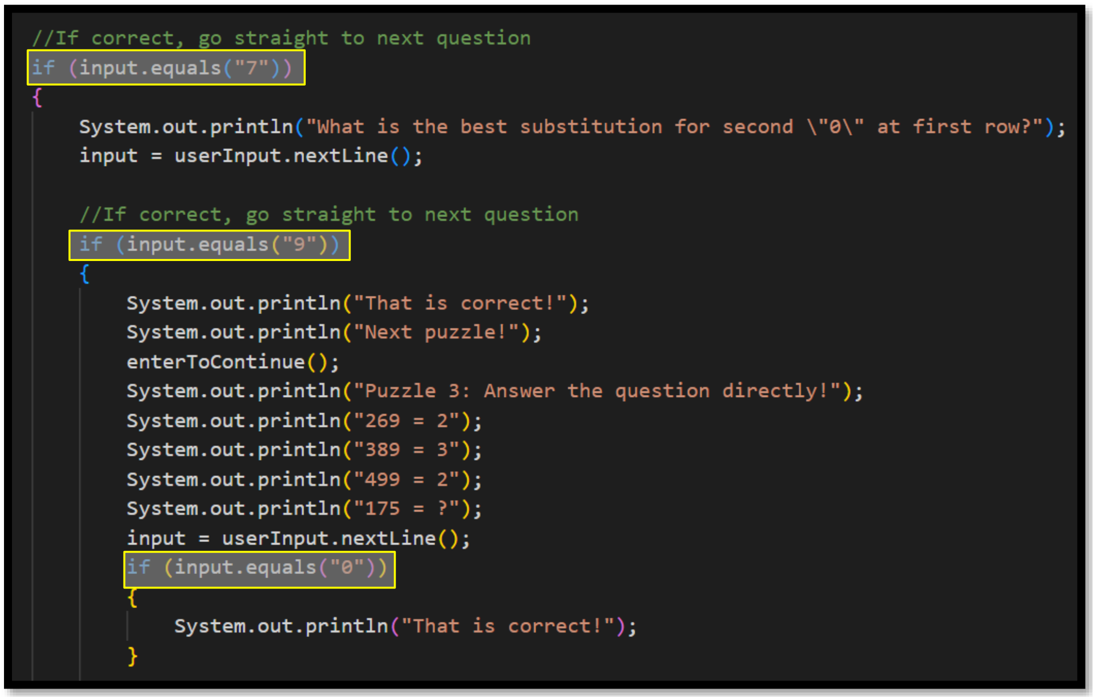
```java
//Recall to the game, find the password
if (input.equals("4790"))
{
    System.out.println("That's correct! The machine opened!");
}

//Incorrect password leads to normal ending
else
{
    System.out.println("That's not correct!");
    enterToContinue();
    System.out.println("Machine: Illegal infiltration! Illegal inflitration!...");
    enterToContinue();
    System.out.println("You have run away from the forest");
    wallaceNormalEnding();
    return;
}
```

### 4.6.3 Nested If statements

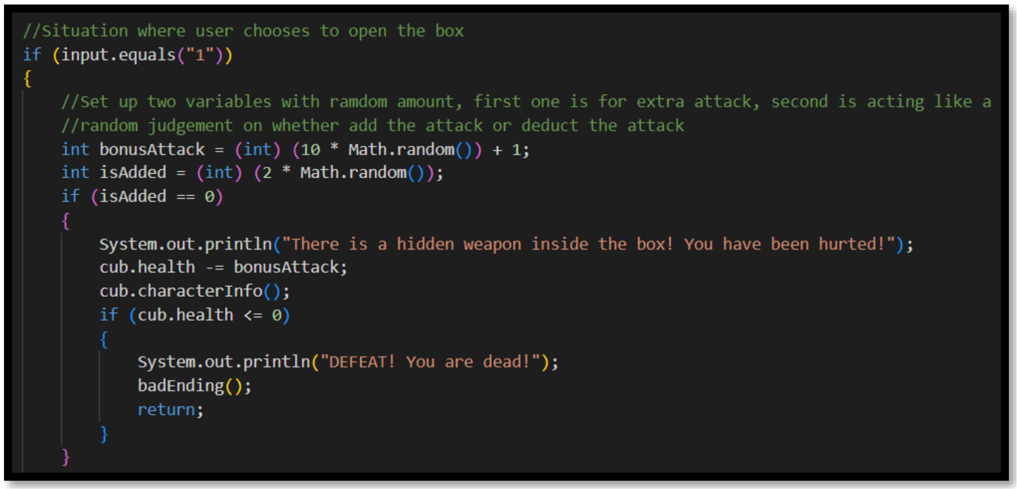
Nested if statements are also used for making different branches of stories, especially for a line which requires a multiple check for user input. For example, the if statement `if (input.equals(“4”))` contains another 3 if statements inside, which makes user need to enter for a correct input for 4 times in order to move next plot.



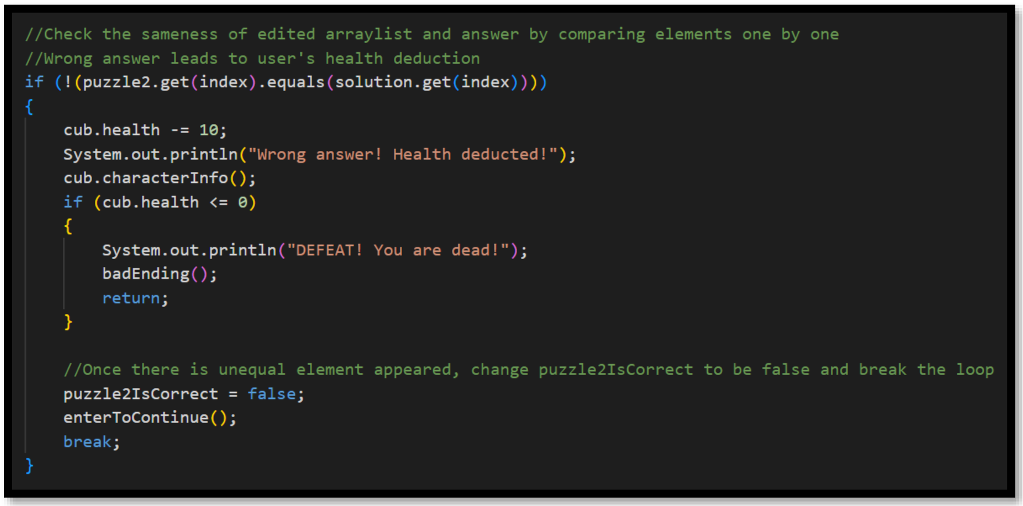
```
if (input.equals("4"))
{
    System.out.println("That is correct!");
    System.out.println("Next Puzzle!");
    enterToContinue();
```



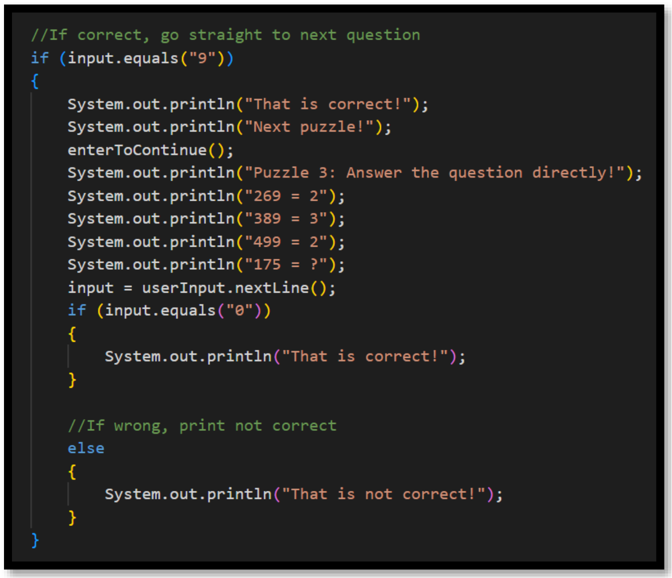
```
//If correct, go straight to next question
if (input.equals("7"))
{
    System.out.println("What is the best substitution for second \"0\" at first row?");
    input = userInput.nextLine();

    //If correct, go straight to next question
    if (input.equals("9"))
    {
        System.out.println("That is correct!");
        System.out.println("Next puzzle!");
        enterToContinue();
        System.out.println("Puzzle 3: Answer the question directly!");
        System.out.println("269 = 2");
        System.out.println("389 = 3");
        System.out.println("499 = 2");
        System.out.println("175 = ?");
        input = userInput.nextLine();
        if (input.equals("0"))
        {
            System.out.println("That is correct!");
        }
```



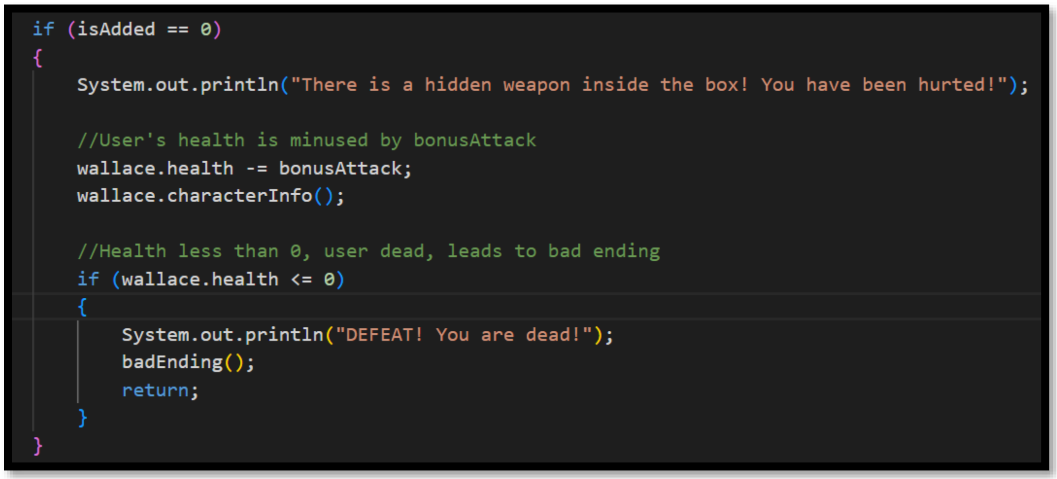
```java
//Situation where user chooses to open the box
if (input.equals("1"))
{
    //Set up two variables with ramdom amount, first one is for extra attack, second is acting like a
    //random judgement on whether add the attack or deduct the attack
    int bonusAttack = (int) (10 * Math.random()) + 1;
    int isAdded = (int) (2 * Math.random());
    if (isAdded == 0)
    {
        System.out.println("There is a hidden weapon inside the box! You have been hurted!");
        cub.health -= bonusAttack;
        cub.characterInfo();
        if (cub.health <= 0)
        {
            System.out.println("DEFEAT! You are dead!");
            badEnding();
            return;
        }
    }
```



```java
//Check the sameness of edited arraylist and answer by comparing elements one by one
//Wrong answer leads to user's health deduction
if (!(puzzle2.get(index).equals(solution.get(index))))
{
    cub.health -= 10;
    System.out.println("Wrong answer! Health deducted!");
    cub.characterInfo();
    if (cub.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        badEnding();
        return;
    }

    //Once there is unequal element appeared, change puzzle2IsCorrect to be false and break the loop
    puzzle2IsCorrect = false;
    enterToContinue();
    break;
}
```



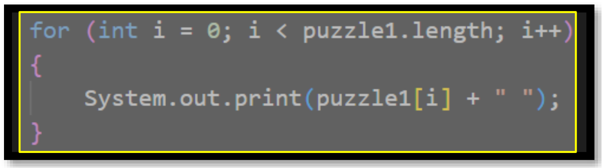
```java
//If correct, go straight to next question
if (input.equals("9"))
{
    System.out.println("That is correct!");
    System.out.println("Next puzzle!");
    enterToContinue();
    System.out.println("Puzzle 3: Answer the question directly!");
    System.out.println("269 = 2");
    System.out.println("389 = 3");
    System.out.println("499 = 2");
    System.out.println("175 = ?");
    input = userInput.nextLine();
    if (input.equals("0"))
    {
        System.out.println("That is correct!");
    }

    //If wrong, print not correct
    else
    {
        System.out.println("That is not correct!");
    }
}
```

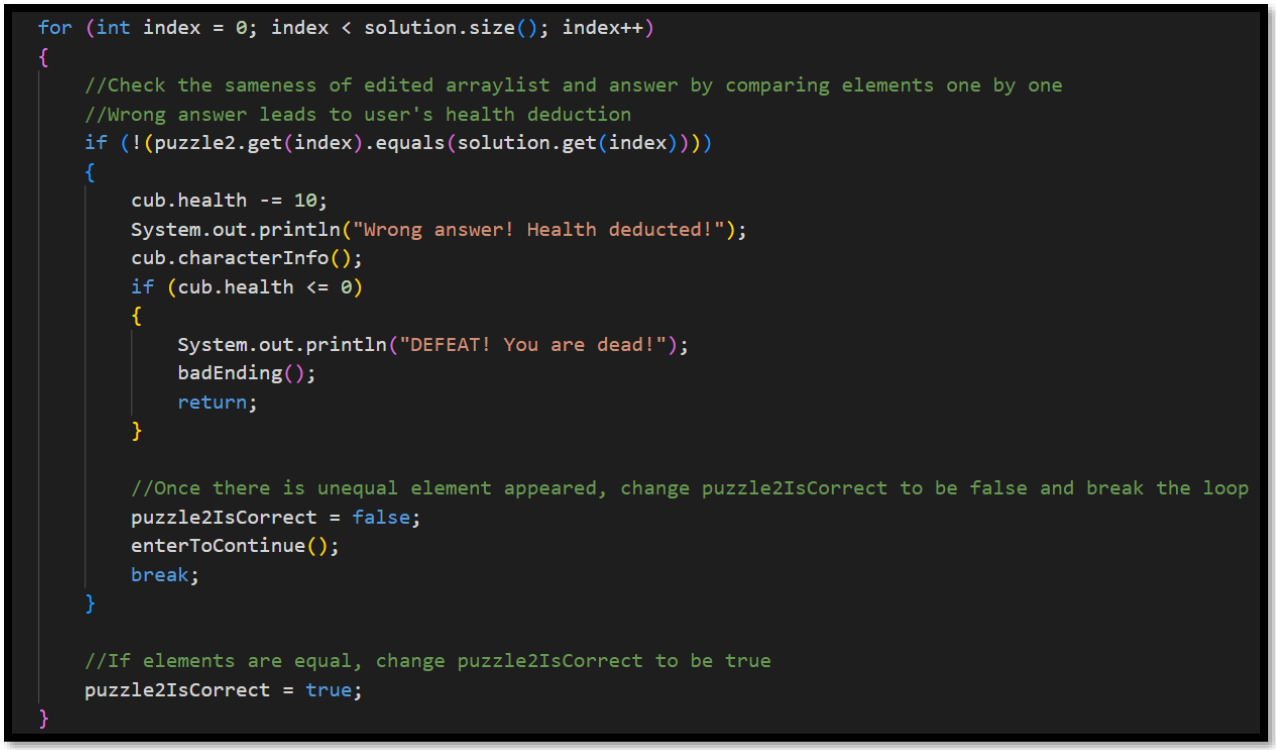

```
if (isAdded == 0)
{
    System.out.println("There is a hidden weapon inside the box! You have been hurted!");

    //User's health is minused by bonusAttack
    wallace.health -= bonusAttack;
    wallace.characterInfo();

    //Health less than 0, user dead, leads to bad ending
    if (wallace.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        badEnding();
        return;
    }
}
```

## *4.7. Loops*

### 4.7.1 For loops

For loops is not frequently used in this program since for game development, it is not possible to know how many times that user needs to be iterated in a specific session. Therefore, these for loops are always applied as an array printer or an array checker (Since array's length is fixed, which makes the times of iteration fixed). For example, the code segment right below has initialized with index number 0, then continue to print the element of array based on index number until it reaches the length of the puzzle. For the output, a whole outlook of array will be displayed.

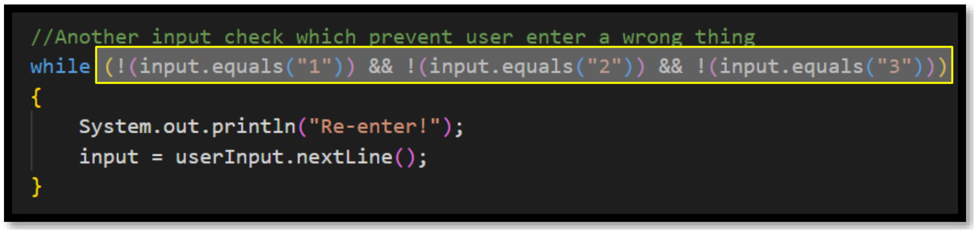

```
for (int i = 0; i < puzzle1.length; i++)
{
    System.out.print(puzzle1[i] + " ");
}
```



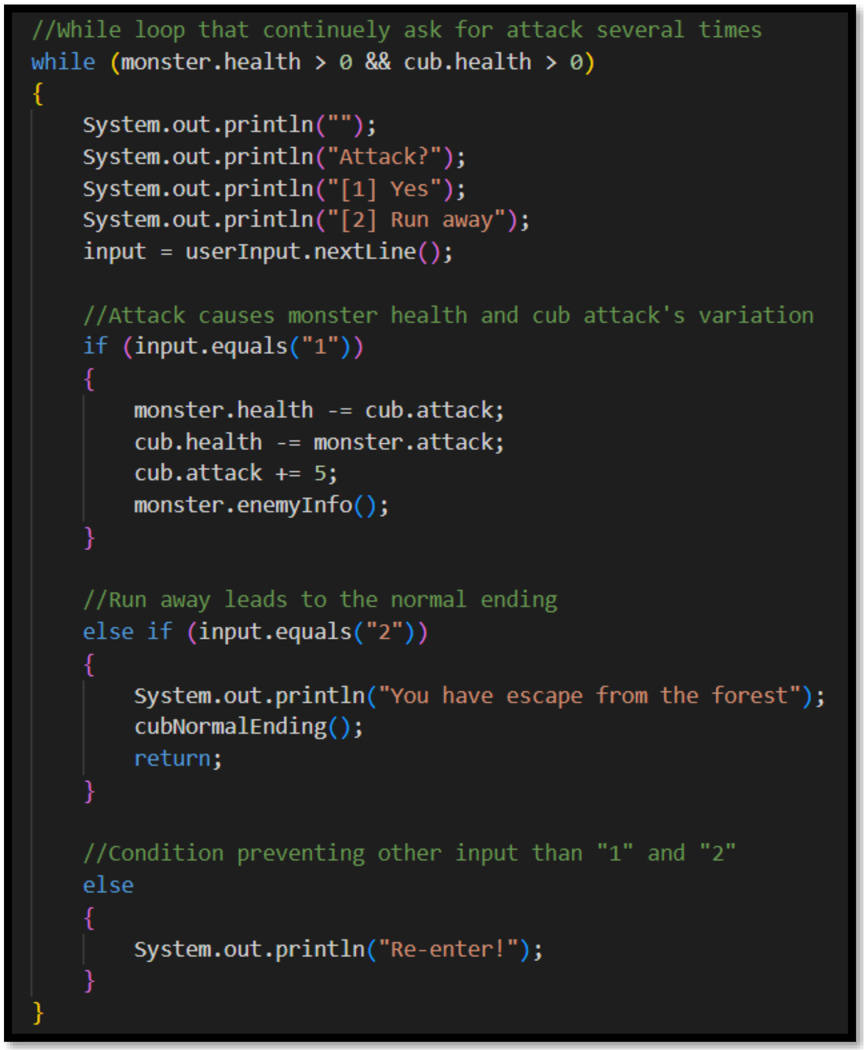
```java
for (int index = 0; index < solution.size(); index++)
{
    //Check the sameness of edited arraylist and answer by comparing elements one by one
    //Wrong answer leads to user's health deduction
    if (!(puzzle2.get(index).equals(solution.get(index))))
    {
        cub.health -= 10;
        System.out.println("Wrong answer! Health deducted!");
        cub.characterInfo();
        if (cub.health <= 0)
        {
            System.out.println("DEFEAT! You are dead!");
            badEnding();
            return;
        }

        //Once there is unequal element appeared, change puzzle2IsCorrect to be false and break the loop
        puzzle2IsCorrect = false;
        enterToContinue();
        break;
    }

    //If elements are equal, change puzzle2IsCorrect to be true
    puzzle2IsCorrect = true;
}
```

### 4.7.2 While Loops

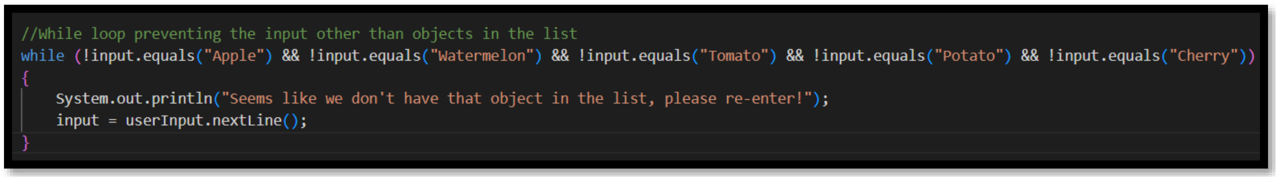
Compared to for loops, while loops are used much frequent, since games possess multiple uncertainties, it is better to have a loop which driven under a specific condition. Therefore, while loops in this program are used to ensure the correctness of user's input, which could let user for re-enters if the input is not satisfied with certain conditions. For example, the while loop showing below has set up a condition `!(input.equals(“1”)) && !(input.equals(“2”)) && !(input.equals(“3”))`, which makes the loop always iterates if `input` is not equals 1, 2 or 3, therefore it will make sure the user input is correct.



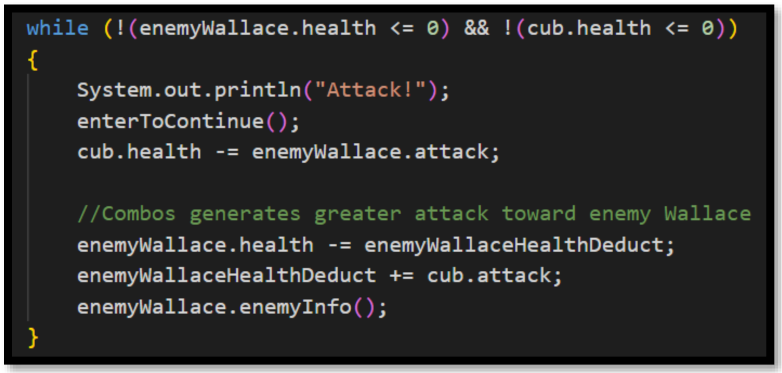
```java
//Another input check which prevent user enter a wrong thing
while (!(input.equals("1")) && !(input.equals("2")) && !(input.equals("3")))
{
    System.out.println("Re-enter!");
    input = userInput.nextLine();
}
```



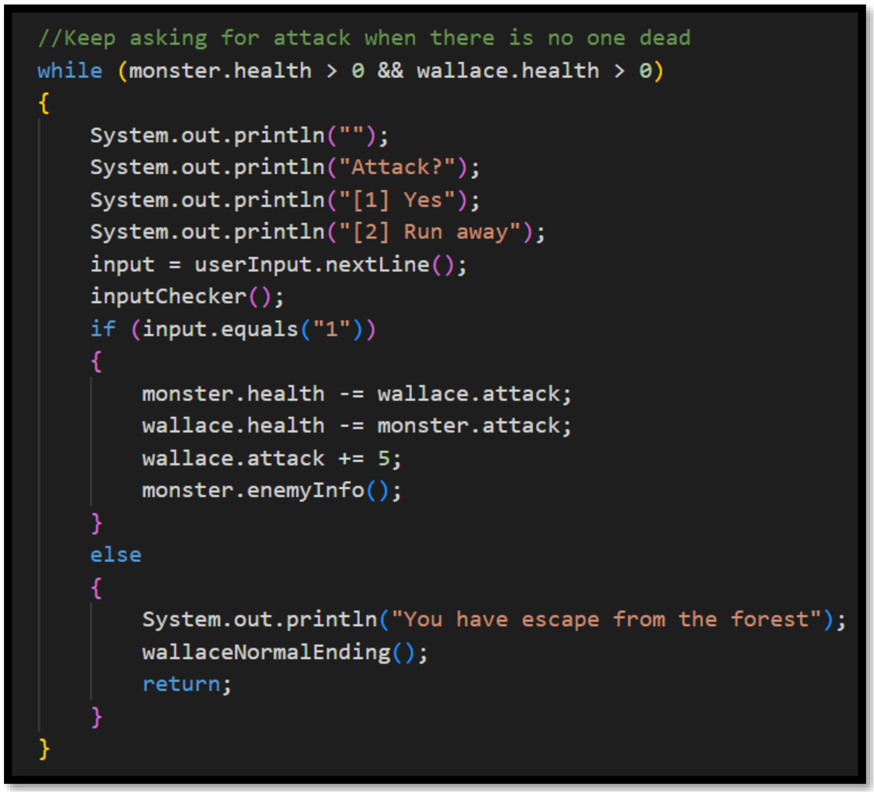
```java
//While loop that continuely ask for attack several times
while (monster.health > 0 && cub.health > 0)
{
    System.out.println("");
    System.out.println("Attack?");
    System.out.println("[1] Yes");
    System.out.println("[2] Run away");
    input = userInput.nextLine();

    //Attack causes monster health and cub attack's variation
    if (input.equals("1"))
    {
        monster.health -= cub.attack;
        cub.health -= monster.attack;
        cub.attack += 5;
        monster.enemyInfo();
    }

    //Run away leads to the normal ending
    else if (input.equals("2"))
    {
        System.out.println("You have escape from the forest");
        cubNormalEnding();
        return;
    }

    //Condition preventing other input than "1" and "2"
    else
    {
        System.out.println("Re-enter!");
    }
}
```



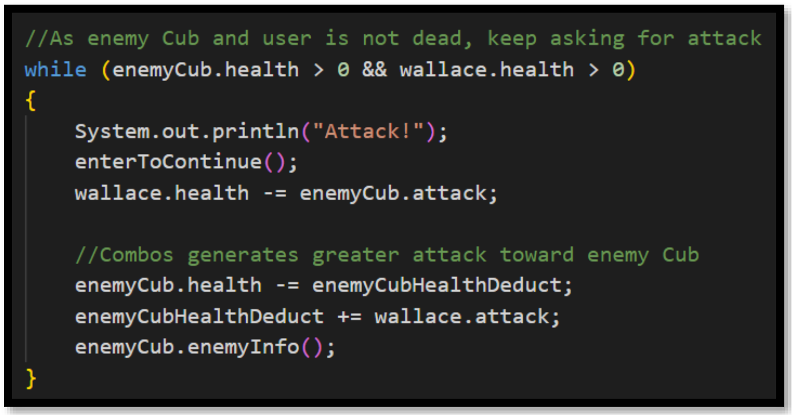
```java
//While loop preventing the input other than objects in the list
while (!input.equals("Apple") && !input.equals("Watermelon") && !input.equals("Tomato") && !input.equals("Potato") && !input.equals("Cherry"))
{
    System.out.println("Seems like we don't have that object in the list, please re-enter!");
    input = userInput.nextLine();
}
```



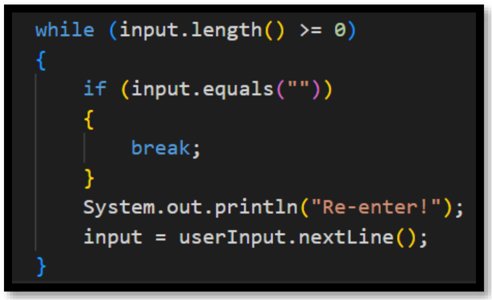
```java
while (!(enemyWallace.health <= 0) && !(cub.health <= 0))
{
    System.out.println("Attack!");
    enterToContinue();
    cub.health -= enemyWallace.attack;

    //Combos generates greater attack toward enemy Wallace
    enemyWallace.health -= enemyWallaceHealthDeduct;
    enemyWallaceHealthDeduct += cub.attack;
    enemyWallace.enemyInfo();
}
```



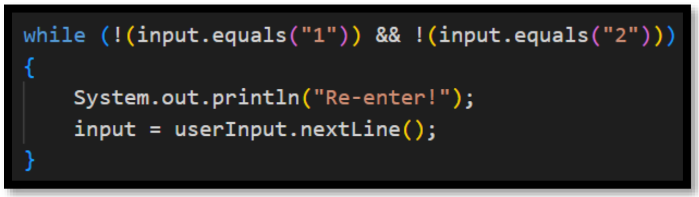
```java
//Keep asking for attack when there is no one dead
while (monster.health > 0 && wallace.health > 0)
{
    System.out.println("");
    System.out.println("Attack?");
    System.out.println("[1] Yes");
    System.out.println("[2] Run away");
    input = userInput.nextLine();
    inputChecker();
    if (input.equals("1"))
    {
        monster.health -= wallace.attack;
        wallace.health -= monster.attack;
        wallace.attack += 5;
        monster.enemyInfo();
    }
    else
    {
        System.out.println("You have escape from the forest");
        wallaceNormalEnding();
        return;
    }
}
```



```java
//As enemy Cub and user is not dead, keep asking for attack
while (enemyCub.health > 0 && wallace.health > 0)
{
    System.out.println("Attack!");
    enterToContinue();
    wallace.health -= enemyCub.attack;

    //Combos generates greater attack toward enemy Cub
    enemyCub.health -= enemyCubHealthDeduct;
    enemyCubHealthDeduct += wallace.attack;
    enemyCub.enemyInfo();
}
```

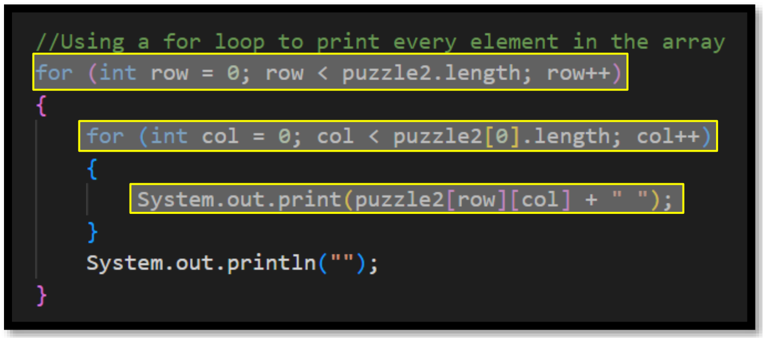


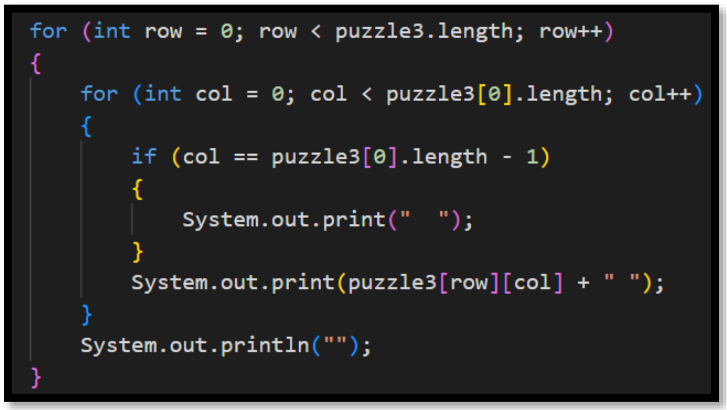
```java
while (input.length() >= 0)
{
    if (input.equals(""))
    {
        break;
    }
    System.out.println("Re-enter!");
    input = userInput.nextLine();
}
```


```java
while (!(input.equals("1")) && !(input.equals("2")))
{
    System.out.println("Re-enter!");
    input = userInput.nextLine();
}
```

### 4.7.3 Nested loops

Nested loops are used for printing 2D-arrays only in the program. For example, the code showing below has applied this function by name the outer for loop variable be `row`, which iterates `puzzle2.length` times (In other words, the number of rows) on printing rows, also name the inner for loop variable be `col`, which iterates `puzzle2[0].length` times (In other words, the number of elements in a row) on printing elements in each row, therefore makes the output will print the whole 2D array.

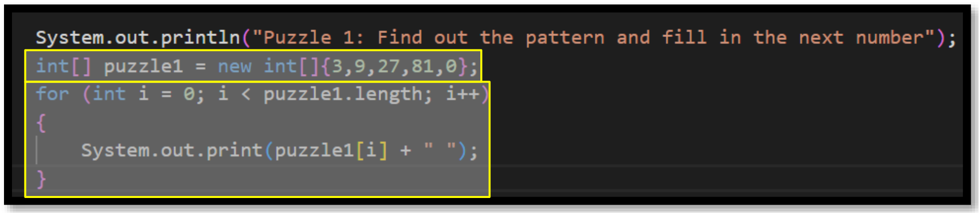

```java
//Using a for loop to print every element in the array
for (int row = 0; row < puzzle2.length; row++)
{
    for (int col = 0; col < puzzle2[0].length; col++)
    {
        System.out.print(puzzle2[row][col] + " ");
    }
    System.out.println("");
}
```


```java
for (int row = 0; row < puzzle3.length; row++)
{
    for (int col = 0; col < puzzle3[0].length; col++)
    {
        if (col == puzzle3[0].length - 1)
        {
            System.out.print("  ");
        }
        System.out.print(puzzle3[row][col] + " ");
    }
    System.out.println("");
}
```

## *4.8 Arrays*

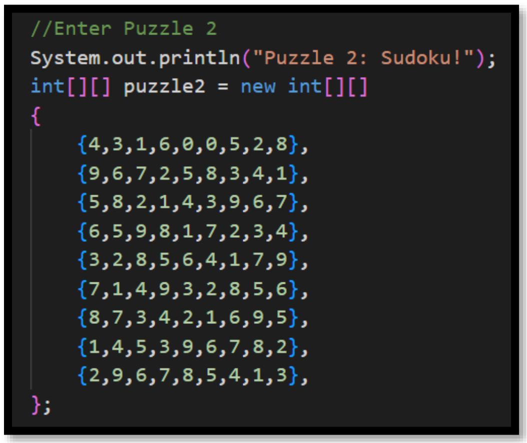
Arrays are mostly used for puzzles, which is the trickiest part during the game, users are required for some math and logic knowledge to pass these puzzles. For example, the 1D array below has been initialized using key word `new`, then every elements have been printed out using for loop in order to let the user answer the question.

1D array:



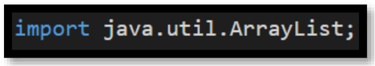
```
System.out.println("Puzzle 1: Find out the pattern and fill in the next number");
int[] puzzle1 = new int[]{3,9,27,81,0};
for (int i = 0; i < puzzle1.length; i++)
{
    System.out.print(puzzle1[i] + " ");
}
```

2D array:

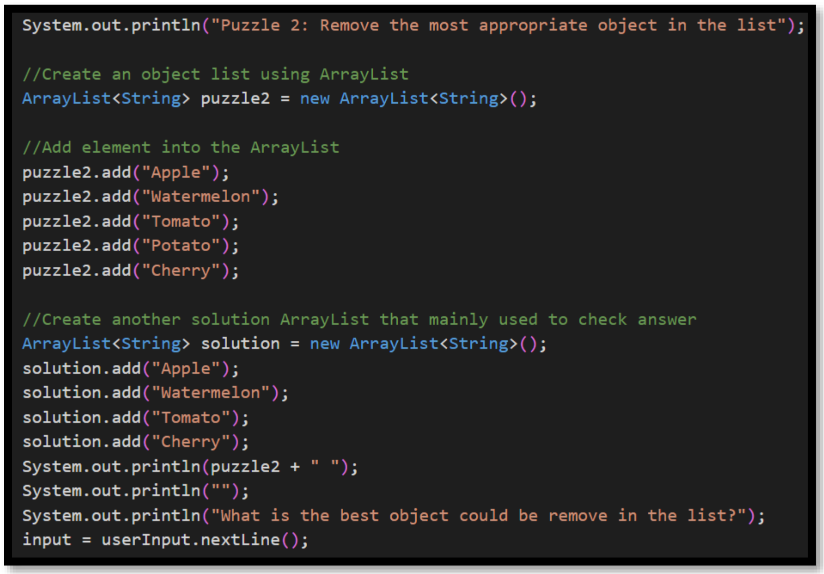


```
//Enter Puzzle 2
System.out.println("Puzzle 2: Sudoku!");
int[][] puzzle2 = new int[][]
{
    {4,3,1,6,0,0,5,2,8},
    {9,6,7,2,5,8,3,4,1},
    {5,8,2,1,4,3,9,6,7},
    {6,5,9,8,1,7,2,3,4},
    {3,2,8,5,6,4,1,7,9},
    {7,1,4,9,3,2,8,5,6},
    {8,7,3,4,2,1,6,9,5},
    {1,4,5,3,9,6,7,8,2},
    {2,9,6,7,8,5,4,1,3},
};
```

Array list:



```
import java.util.ArrayList;
```

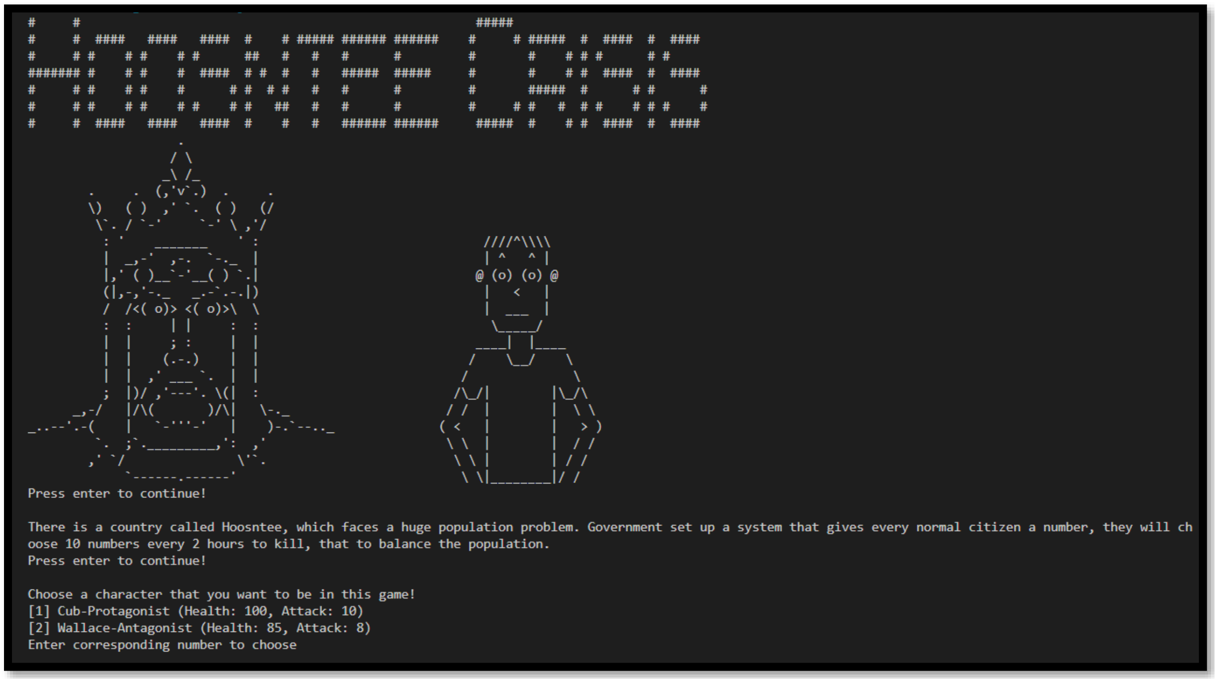


```
System.out.println("Puzzle 2: Remove the most appropriate object in the list");

//Create an object list using ArrayList
ArrayList<String> puzzle2 = new ArrayList<String>();

//Add element into the ArrayList
puzzle2.add("Apple");
puzzle2.add("Watermelon");
puzzle2.add("Tomato");
puzzle2.add("Potato");
puzzle2.add("Cherry");

//Create another solution ArrayList that mainly used to check answer
ArrayList<String> solution = new ArrayList<String>();
solution.add("Apple");
solution.add("Watermelon");
solution.add("Tomato");
solution.add("Cherry");
System.out.println(puzzle2 + " ");
System.out.println("");
System.out.println("What is the best object could be remove in the list?");
input = userInput.nextLine();
```

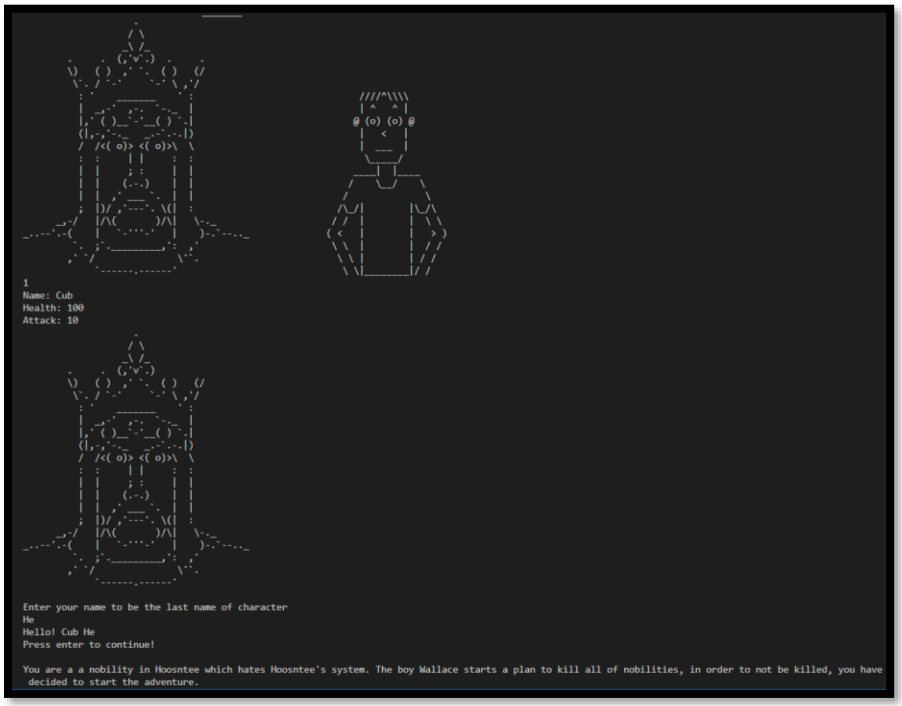
## 5. Testing, Issues and Efficiency Development

### *5.1 Testing*

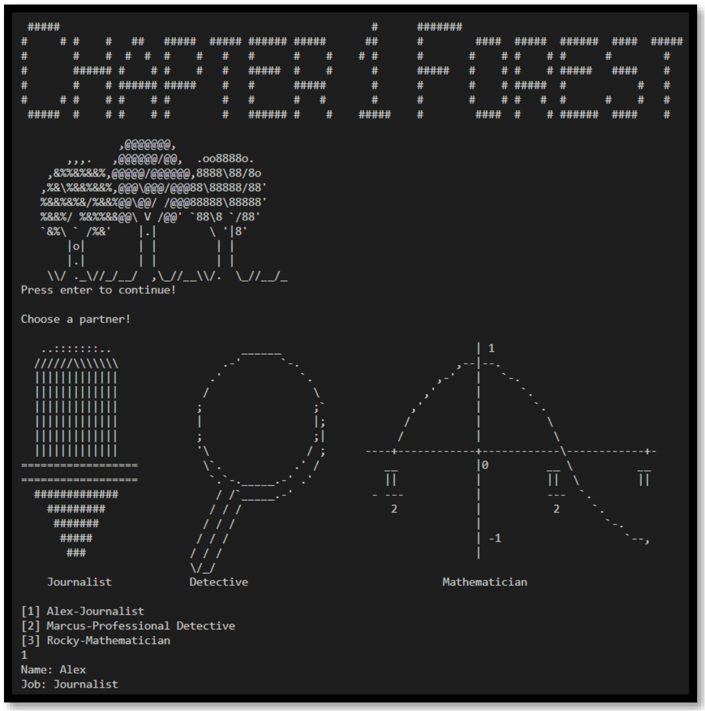
Testing is the most significant part for a game development, first I am going to show an output for my adventure game *Hoosntee Crisis* (Cub's true endings).



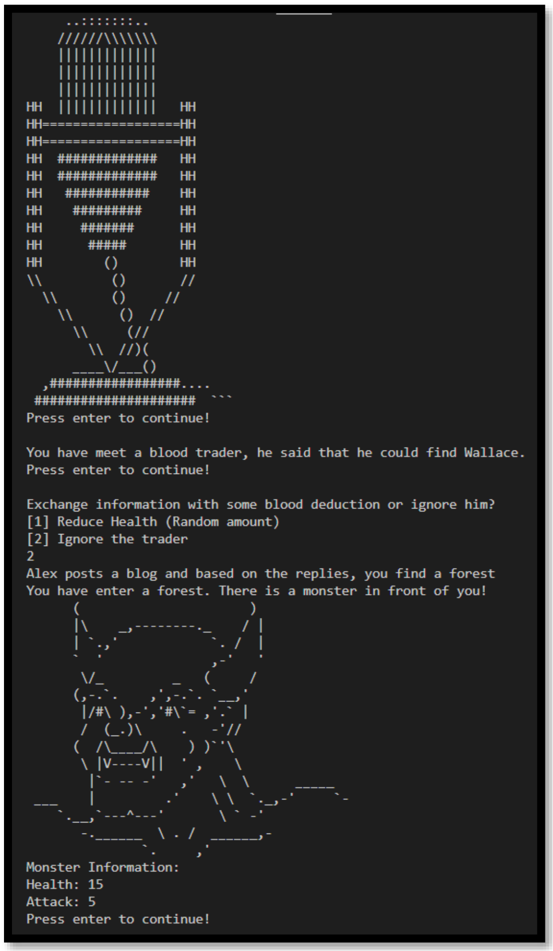
```
Press enter to continue!

There is a country called Hoosntee, which faces a huge population problem. Government set up a system that gives every normal citizen a number, they will ch
oose 10 numbers every 2 hours to kill, that to balance the population.
Press enter to continue!

Choose a character that you want to be in this game!
[1] Cub-Protagonist (Health: 100, Attack: 10)
[2] Wallace-Antagonist (Health: 85, Attack: 8)
Enter corresponding number to choose
```



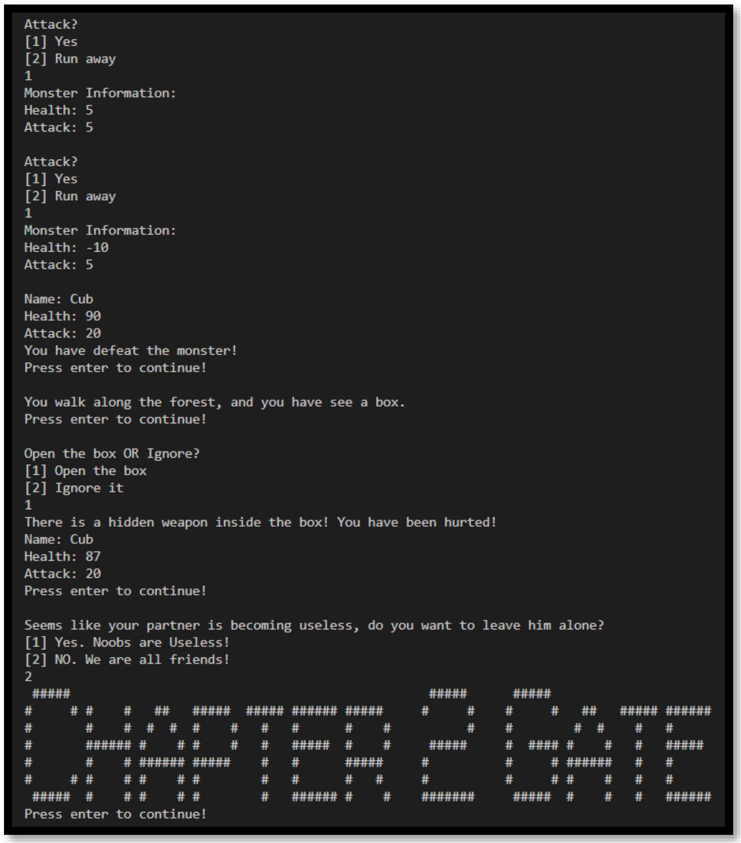
```
1
Name: Cub
Health: 100
Attack: 10

Enter your name to be the last name of character
He
Hello! Cub He
Press enter to continue!

You are a a nobility in Hoosntee which hates Hoosntee's system. The boy Wallace starts a plan to kill all of nobilities, in order to not be killed, you have
 decided to start the adventure.
```



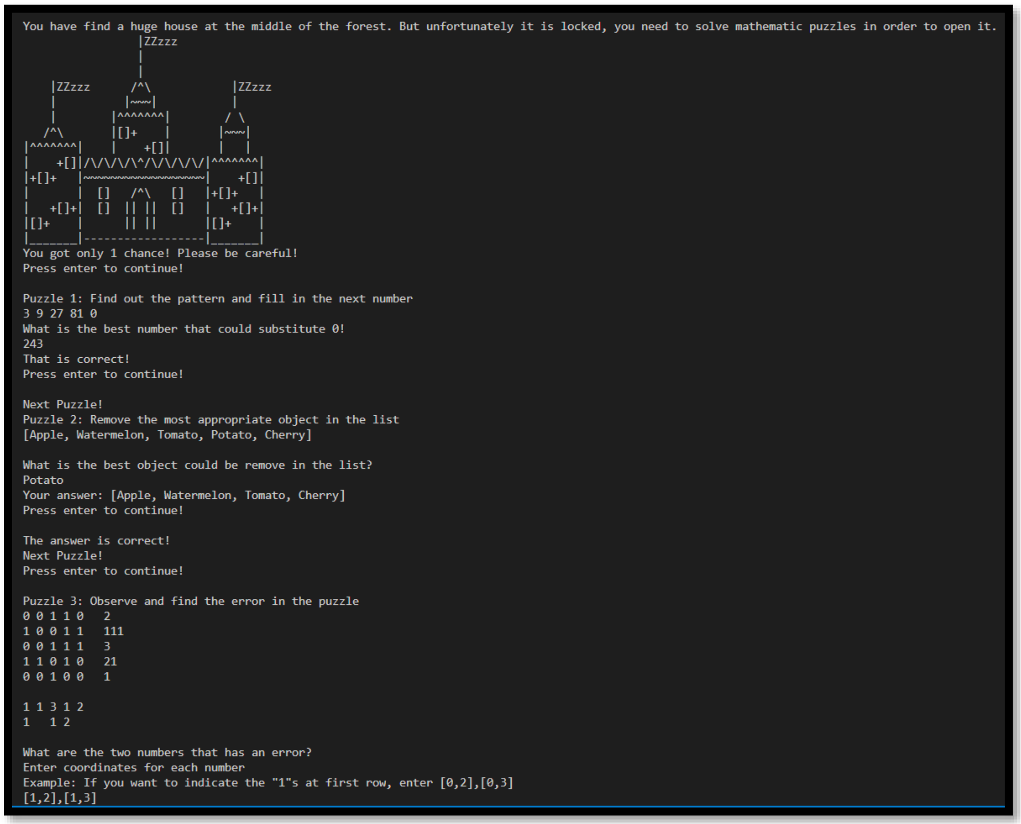
```
Press enter to continue!

Choose a partner!

    Journalist          Detective                    Mathematician

[1] Alex-Journalist
[2] Marcus-Professional Detective
[3] Rocky-Mathematician
1
Name: Alex
Job: Journalist
```



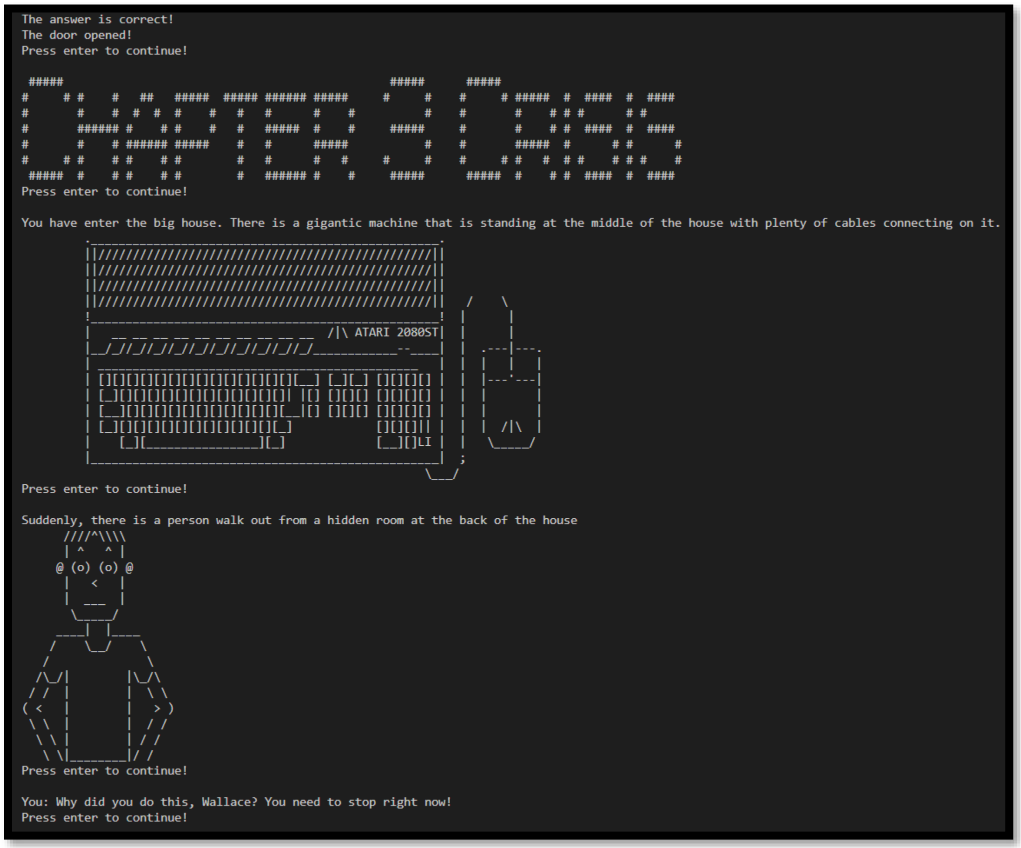
```
Press enter to continue!

You have meet a blood trader, he said that he could find Wallace.
Press enter to continue!

Exchange information with some blood deduction or ignore him?
[1] Reduce Health (Random amount)
[2] Ignore the trader
2
Alex posts a blog and based on the replies, you find a forest
You have enter a forest. There is a monster in front of you!

Monster Information:
Health: 15
Attack: 5
Press enter to continue!
```



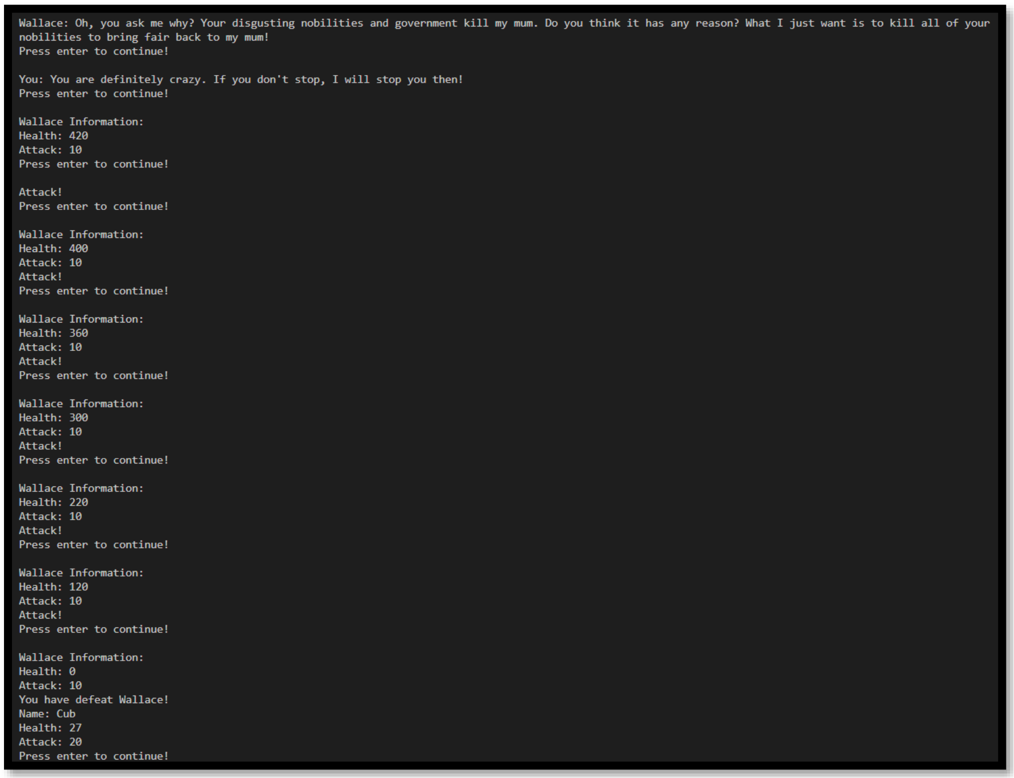
```
Attack?
[1] Yes
[2] Run away
1
Monster Information:
Health: 5
Attack: 5

Attack?
[1] Yes
[2] Run away
1
Monster Information:
Health: -10
Attack: 5

Name: Cub
Health: 90
Attack: 20
You have defeat the monster!
Press enter to continue!

You walk along the forest, and you have see a box.
Press enter to continue!

Open the box OR Ignore?
[1] Open the box
[2] Ignore it
1
There is a hidden weapon inside the box! You have been hurted!
Name: Cub
Health: 87
Attack: 20
Press enter to continue!

Seems like your partner is becoming useless, do you want to leave him alone?
[1] Yes. Noobs are Useless!
[2] NO. We are all friends!
2
 #####                                               #####      #####
#     # #    #   ##   #####  ##### ###### #####     #     #    #     #   ##   ##### ######
#       #    #  #  #  #    #   #   #      #    #          #    #        #  #    #   #
#       ###### #    # #    #   #   #####  #    #     #####     #  #### #    #   #   #####
#       #    # ###### #####    #   #      #####     #          #     # ######   #   #
#     # #    # #    # #        #   #      #   #     #          #     # #    #   #   #
 #####  #    # #    # #        #   ###### #    #    #######     #####  #    #   #   ######
Press enter to continue!
```



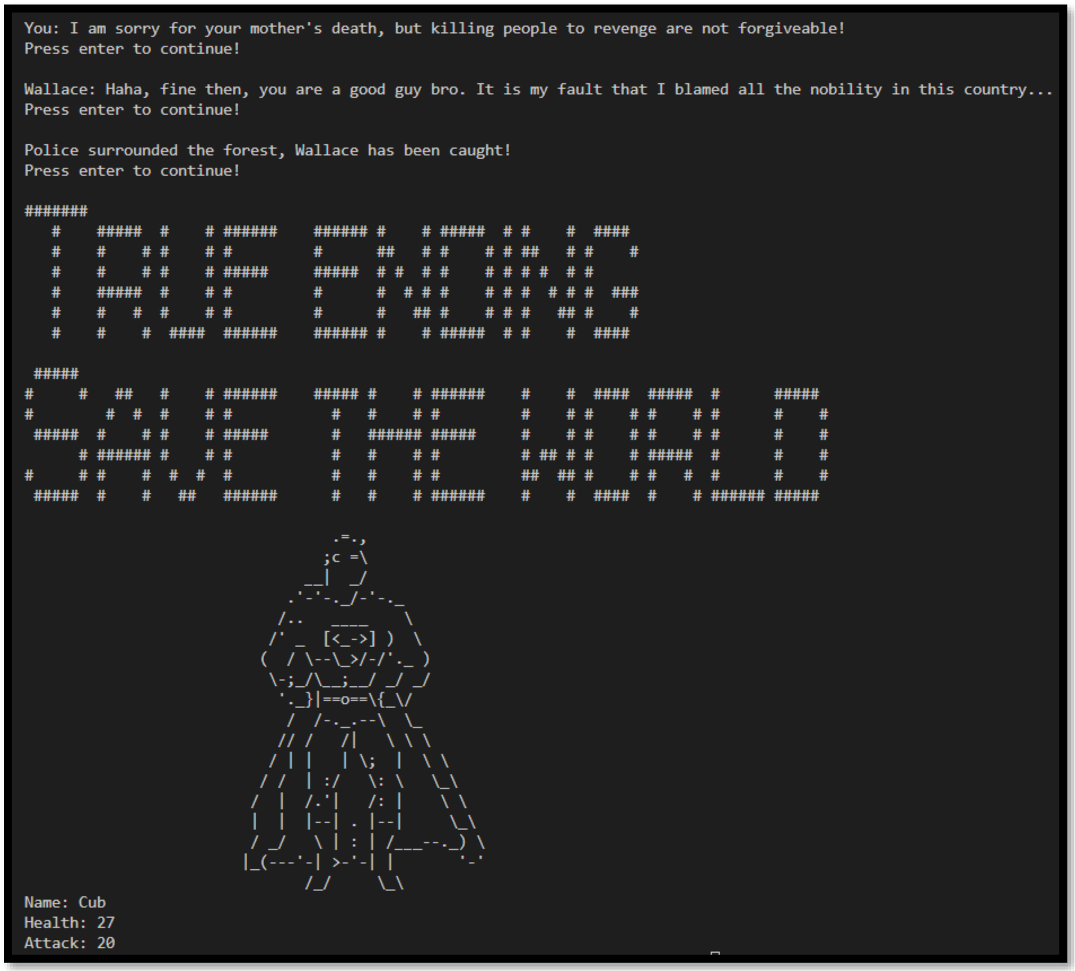
```
You have find a huge house at the middle of the forest. But unfortunately it is locked, you need to solve mathematic puzzles in order to open it.
                 |ZZzzz
                 |
                 |
    |ZZzzz      /^\           |ZZzzz
    |          |~~~|           |
    |        |^^^^^^^|        / \
   /^\       |[]+    |       |~~~|
|^^^^^^^|    |    +[]|       |   |
|    +[]|/\/\/\/\^/\/\/\/\/|^^^^^^^|
|+[]+   |~~~~~~~~~~~~~~~~~~|    +[]|
|       |  []   /^\   []   |+[]+   |
|   +[]+|  []  || ||  []   |   +[]+|
|[]+    |      || ||       |[]+    |
|_______|------------------|_______|
You got only 1 chance! Please be careful!
Press enter to continue!

Puzzle 1: Find out the pattern and fill in the next number
3 9 27 81 0
What is the best number that could substitute 0!
243
That is correct!
Press enter to continue!

Next Puzzle!
Puzzle 2: Remove the most appropriate object in the list
[Apple, Watermelon, Tomato, Potato, Cherry]

What is the best object could be remove in the list?
Potato
Your answer: [Apple, Watermelon, Tomato, Cherry]
Press enter to continue!

The answer is correct!
Next Puzzle!
Press enter to continue!

Puzzle 3: Observe and find the error in the puzzle
0 0 1 1 0   2
1 0 0 1 1   111
0 0 1 1 1   3
1 1 0 1 0   21
0 0 1 0 0   1

1 1 3 1 2
1   1 2

What are the two numbers that has an error?
Enter coordinates for each number
Example: If you want to indicate the "1"s at first row, enter [0,2],[0,3]
[1,2],[1,3]
```



```
The answer is correct!
The door opened!
Press enter to continue!

 #####                                                #####      #####
#     # #    #   ##   #####  ##### ###### #####      #     #    #     # #####  #  ####  #  ####
#       #    #  #  #  #    #   #   #      #    #           #    #       #    # # #      # #
#       ###### #    # #    #   #   #####  #    #      #####     #       #    # #  ####  #  ####
#       #    # ###### #####    #   #      #####            #    #       #####  #      # #      #
#     # #    # #    # #        #   #      #    #     #     #    #     # #   #  # #    # # #    #
 #####  #    # #    # #        #   ###### #    #      #####      #####  #    # #  ####  #  ####
Press enter to continue!

You have enter the big house. There is a gigantic machine that is standing at the middle of the house with plenty of cables connecting on it.
          .______________________________________________________.
          ||/////////////////////////////////////////////////////||
          ||/////////////////////////////////////////////////////||
          ||/////////////////////////////////////////////////////||
          ||/////////////////////////////////////////////////////||   /    \
          !______________________________________________________!  |      |
          |    __ __ __ __ __ __ __ __ __ __   /|\ ATARI 2080ST|  |      |
          |__/_//_//_//_//_//_//_//_//_//_/____________--_____|  |  .---|---.
          |  ____________________________________________      |  |  |   |   |
          | [][][][][][][][][][][][][][__] [_][_] [][][][] |  |  |---'---|
          | [_][][][][][][][][][][][][]| |[] [][][] [][][][] |  |  |       |
          | [__][][][][][][][][][][][__|[] [][][] [][][][] |  |  |       |
          | [_][][][][][][][][][][][_]            [][][]|| |  |  |  /|\  |
          |    [_][________________][_]            [__][]LI |  |   \_____/
          |______________________________________________________|  ;
                                                            \___/
Press enter to continue!

Suddenly, there is a person walk out from a hidden room at the back of the house
      ////^\\\\
      | ^   ^ |
     @ (o) (o) @
      |   <   |
      |  ___  |
       \_____/
     ____|  |____
    /    \__/    \
   /              \
  /\_/|        |\_/\
 / /  |        |  \ \
( <   |        |   > )
 \ \  |        |  / /
  \ \ |        | / /
   \ \|________|/ /
Press enter to continue!

You: Why did you do this, Wallace? You need to stop right now!
Press enter to continue!
```



```
Wallace: Oh, you ask me why? Your disgusting nobilities and government kill my mum. Do you think it has any reason? What I just want is to kill all of your
nobilities to bring fair back to my mum!
Press enter to continue!

You: You are definitely crazy. If you don't stop, I will stop you then!
Press enter to continue!

Wallace Information:
Health: 420
Attack: 10
Press enter to continue!

Attack!
Press enter to continue!

Wallace Information:
Health: 400
Attack: 10
Attack!
Press enter to continue!

Wallace Information:
Health: 360
Attack: 10
Attack!
Press enter to continue!

Wallace Information:
Health: 300
Attack: 10
Attack!
Press enter to continue!

Wallace Information:
Health: 220
Attack: 10
Attack!
Press enter to continue!

Wallace Information:
Health: 120
Attack: 10
Attack!
Press enter to continue!

Wallace Information:
Health: 0
Attack: 10
You have defeat Wallace!
Name: Cub
Health: 27
Attack: 20
Press enter to continue!
```

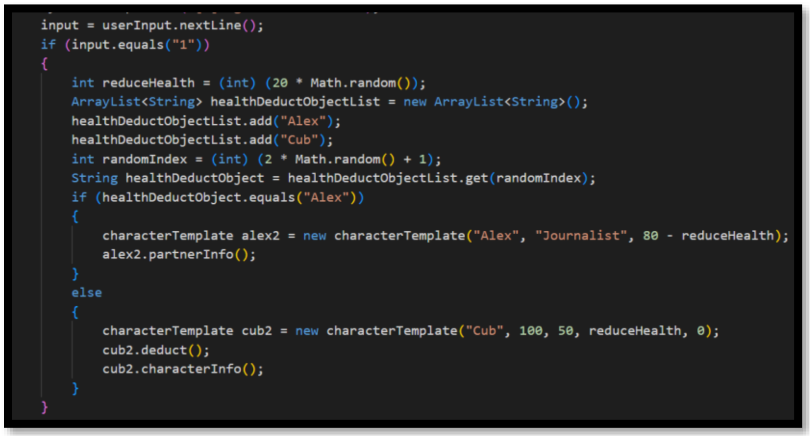


```
You: I am sorry for your mother's death, but killing people to revenge are not forgiveable!
Press enter to continue!

Wallace: Haha, fine then, you are a good guy bro. It is my fault that I blamed all the nobility in this country...
Press enter to continue!

Police surrounded the forest, Wallace has been caught!
Press enter to continue!

#######
   #    #####  #    # ######    ###### #    # #####  # #    #  ####
   #    #    # #    # #         #      ##   # #    # # ##   # #    #
   #    #    # #    # #####     #####  # #  # #    # # # #  # #
   #    #####  #    # #         #      #  # # #    # # #  # # #  ###
   #    #   #  #    # #         #      #   ## #    # # #   ## #    #
   #    #    #  ####  ######    ###### #    # #####  # #    #  ####

 #####
#     #   ##   #    # ######    ##### #    # ######    #    #  ####  #####  #      #####
#        #  #  #    # #           #   #    # #         #    # #    # #    # #      #    #
 #####  #    # #    # #####       #   ###### #####     #    # #    # #    # #      #    #
      # ###### #    # #           #   #    # #         # ## # #    # #####  #      #    #
#     # #    #  #  #  #           #   #    # #         ##  ## #    # #   #  #      #    #
 #####  #    #   ##   ######      #   #    # ######    #    #  ####  #    # ###### #####

                                  .=.,
                                 ;c =\
                               __|  _/
                             .'-'-._/-'-._
                            /..   ____    \
                           /' _  [<_->] )  \
                          (  / \--\_>/-/'._ )
                           \-;_/\__;__/ _/ _/
                            '._}|==o==\{_\/
                             /  /-._.--\  \_
                            // /   /|   \ \ \
                           / | |   | \;  |  \ \
                          / /  | :/   \: \   \_\
                         /  |  /.'|   /: |    \ \
                         |  |  |--| . |--|     \_\
                         / _/   \ | : | /___--._) \
                        |_(---'-| >-'-| |       '-'
                               /_/     \_\

Name: Cub
Health: 27
Attack: 20
```

## 5.2 Issues

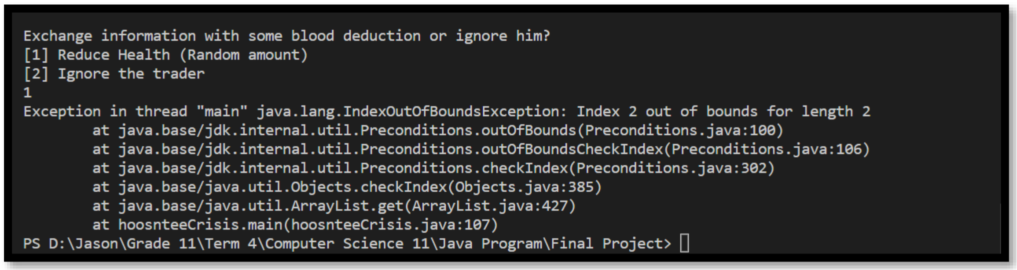
During the process of coding, I have got multiple issues. In this section, I am going to mainly talk about these issues with the way that how did I solved it.

### Issue 1: Index out of bound



```java
input = userInput.nextLine();
if (input.equals("1"))
{
    int reduceHealth = (int) (20 * Math.random());
    ArrayList<String> healthDeductObjectList = new ArrayList<String>();
    healthDeductObjectList.add("Alex");
    healthDeductObjectList.add("Cub");
    int randomIndex = (int) (2 * Math.random() + 1);
    String healthDeductObject = healthDeductObjectList.get(randomIndex);
    if (healthDeductObject.equals("Alex"))
    {
        characterTemplate alex2 = new characterTemplate("Alex", "Journalist", 80 - reduceHealth);
        alex2.partnerInfo();
    }
    else
    {
        characterTemplate cub2 = new characterTemplate("Cub", 100, 50, reduceHealth, 0);
        cub2.deduct();
        cub2.characterInfo();
    }
}
```

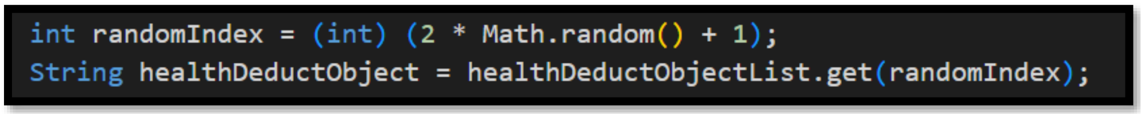
*Original Output:*



```
Exchange information with some blood deduction or ignore him?
[1] Reduce Health (Random amount)
[2] Ignore the trader
1
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 2 out of bounds for length 2
        at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:100)
        at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:106)
        at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:302)
        at java.base/java.util.Objects.checkIndex(Objects.java:385)
        at java.base/java.util.ArrayList.get(ArrayList.java:427)
        at hoosnteeCrisis.main(hoosnteeCrisis.java:107)
PS D:\Jason\Grade 11\Term 4\Computer Science 11\Java Program\Final Project>
```

*What is the problem?*

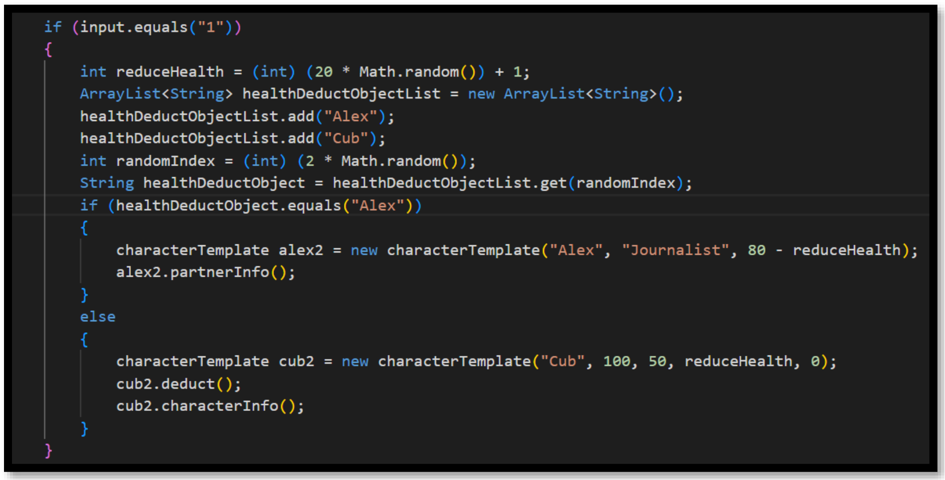
During the implementation of the program, an `IndexOutOfBoundException` has been thrown, which means there is a problem on accessing the array: the length of the array is smaller than the index number.



```
int randomIndex = (int) (2 * Math.random() + 1);
String healthDeductObject = healthDeductObjectList.get(randomIndex);
```

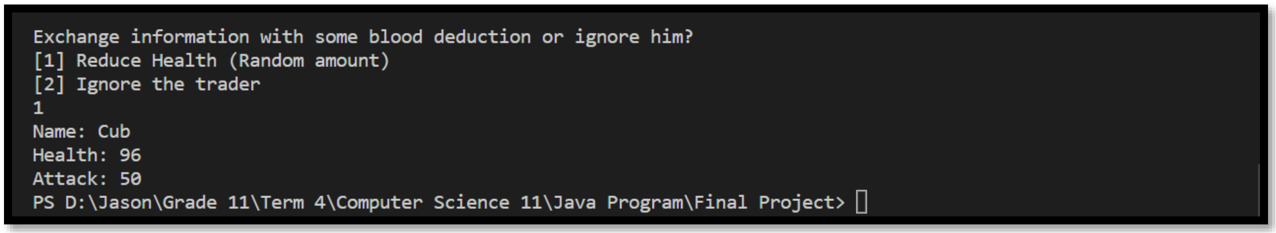
*Solution*

Change the value of the variable `randomIndex` to be `(int) (2 * Math.random)`, which makes random numbers generated in the bound of 0 and 1 (In later stage, this piece of code is already deleted due to the change of plot, but it is still valuable to be performed in this section).



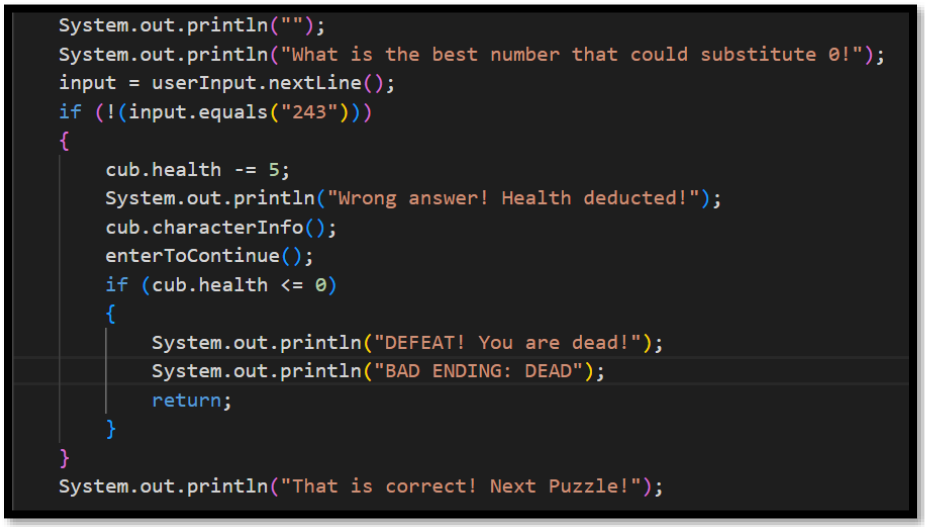
```
if (input.equals("1"))
{
    int reduceHealth = (int) (20 * Math.random()) + 1;
    ArrayList<String> healthDeductObjectList = new ArrayList<String>();
    healthDeductObjectList.add("Alex");
    healthDeductObjectList.add("Cub");
    int randomIndex = (int) (2 * Math.random());
    String healthDeductObject = healthDeductObjectList.get(randomIndex);
    if (healthDeductObject.equals("Alex"))
    {
        characterTemplate alex2 = new characterTemplate("Alex", "Journalist", 80 - reduceHealth);
        alex2.partnerInfo();
    }
    else
    {
        characterTemplate cub2 = new characterTemplate("Cub", 100, 50, reduceHealth, 0);
        cub2.deduct();
        cub2.characterInfo();
    }
}
```

*Output after the change:*



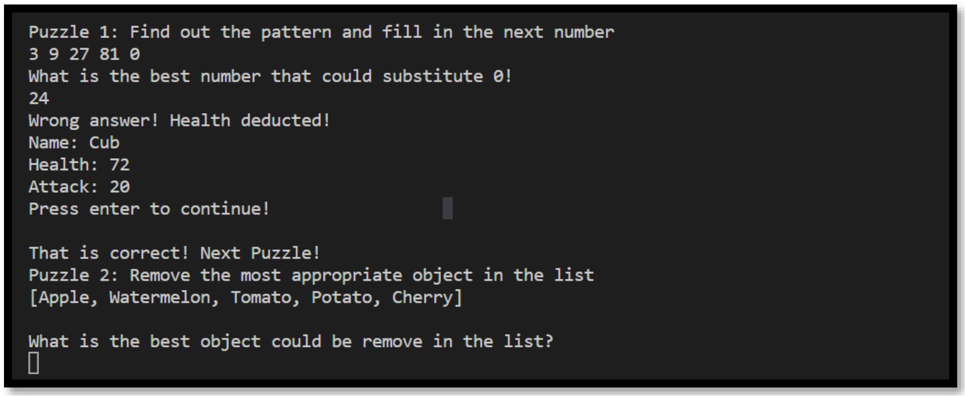
```
Exchange information with some blood deduction or ignore him?
[1] Reduce Health (Random amount)
[2] Ignore the trader
1
Name: Cub
Health: 96
Attack: 50
PS D:\Jason\Grade 11\Term 4\Computer Science 11\Java Program\Final Project> 
```

## Issue 2: Logic Error



```java
System.out.println("");
System.out.println("What is the best number that could substitute 0!");
input = userInput.nextLine();
if (!(input.equals("243")))
{
    cub.health -= 5;
    System.out.println("Wrong answer! Health deducted!");
    cub.characterInfo();
    enterToContinue();
    if (cub.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        System.out.println("BAD ENDING: DEAD");
        return;
    }
}
System.out.println("That is correct! Next Puzzle!");
```

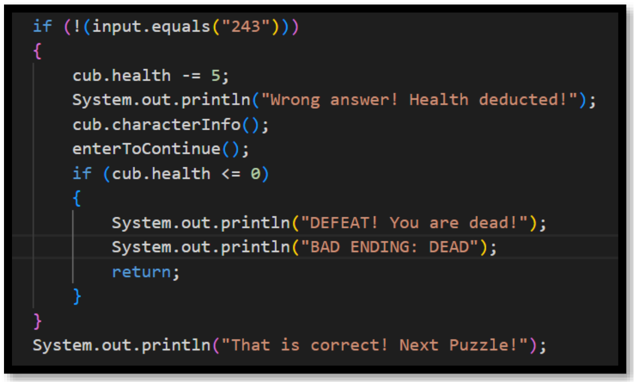
*Original Output:*



```
Puzzle 1: Find out the pattern and fill in the next number
3 9 27 81 0
What is the best number that could substitute 0!
24
Wrong answer! Health deducted!
Name: Cub
Health: 72
Attack: 20
Press enter to continue!

That is correct! Next Puzzle!
Puzzle 2: Remove the most appropriate object in the list
[Apple, Watermelon, Tomato, Potato, Cherry]

What is the best object could be remove in the list?
```

*What is the problem?*

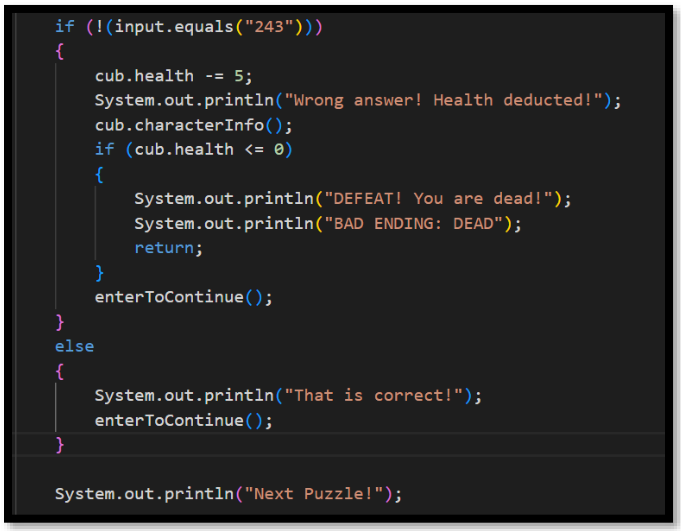
Statement "That is correct! Next Puzzle!" is printed after the wrong answer is given, which means as the if statement is implemented, it directly goes down to the print statement at the bottom, which caused the logic error.



```
if (!(input.equals("243")))
{
    cub.health -= 5;
    System.out.println("Wrong answer! Health deducted!");
    cub.characterInfo();
    enterToContinue();
    if (cub.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        System.out.println("BAD ENDING: DEAD");
        return;
    }
}
System.out.println("That is correct! Next Puzzle!");
```

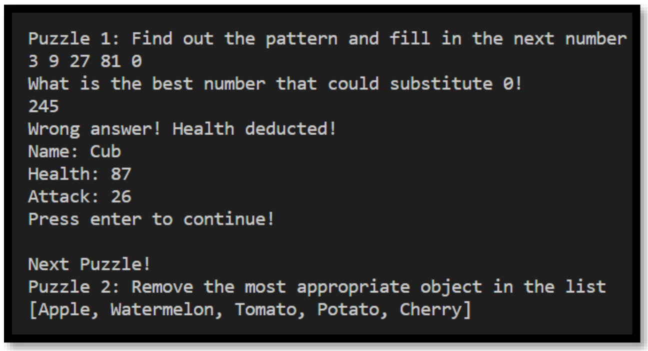
*Solution*

Add an else statement for the correct situation, which establish a correct logic that wrong answers lead to health deduction and correct answer proceed directly.



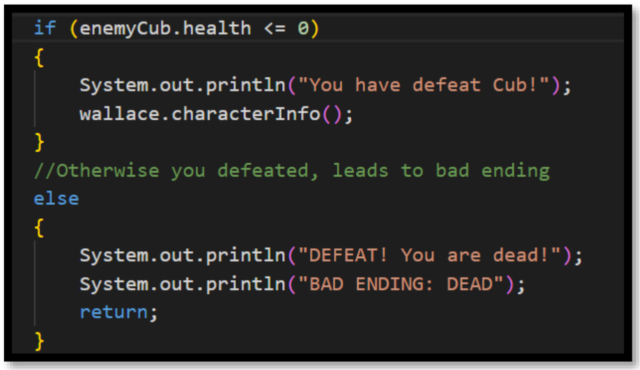
```
if (!(input.equals("243")))
{
    cub.health -= 5;
    System.out.println("Wrong answer! Health deducted!");
    cub.characterInfo();
    if (cub.health <= 0)
    {
        System.out.println("DEFEAT! You are dead!");
        System.out.println("BAD ENDING: DEAD");
        return;
    }
    enterToContinue();
}
else
{
    System.out.println("That is correct!");
    enterToContinue();
}

System.out.println("Next Puzzle!");
```

*Output after the change:*



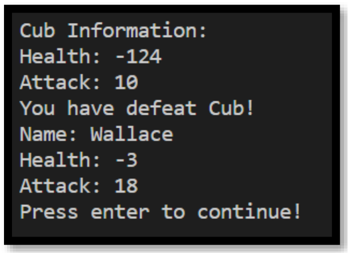
```
Puzzle 1: Find out the pattern and fill in the next number
3 9 27 81 0
What is the best number that could substitute 0!
245
Wrong answer! Health deducted!
Name: Cub
Health: 87
Attack: 26
Press enter to continue!

Next Puzzle!
Puzzle 2: Remove the most appropriate object in the list
[Apple, Watermelon, Tomato, Potato, Cherry]
```

**Issue 3: Condition Error**



```
if (enemyCub.health <= 0)
{
    System.out.println("You have defeat Cub!");
    wallace.characterInfo();
}
//Otherwise you defeated, leads to bad ending
else
{
    System.out.println("DEFEAT! You are dead!");
    System.out.println("BAD ENDING: DEAD");
    return;
}
```

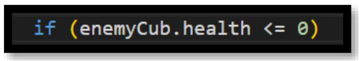
*Original Output:*



```
Cub Information:
Health: -124
Attack: 10
You have defeat Cub!
Name: Wallace
Health: -3
Attack: 18
Press enter to continue!
```

*What is the problem?*

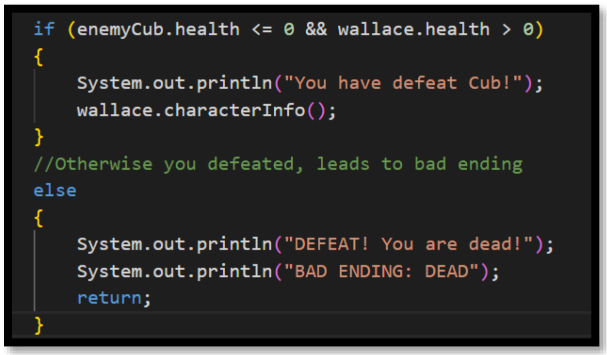
The set up of the condition `enemyCub.health <= 0` does not consider the situation where both user and the boss dead at the last moment, which makes the output extremely weird.



```
if (enemyCub.health <= 0)
```

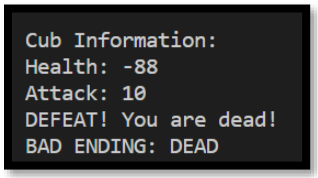
*Solution*

Change the condition in if statement to `enemyCub.health <= 0 && Wallace.health > 0)`, which includes the situation for both boss and user dead at last moment.



```
if (enemyCub.health <= 0 && wallace.health > 0)
{
    System.out.println("You have defeat Cub!");
    wallace.characterInfo();
}
//Otherwise you defeated, leads to bad ending
else
{
    System.out.println("DEFEAT! You are dead!");
    System.out.println("BAD ENDING: DEAD");
    return;
}
```

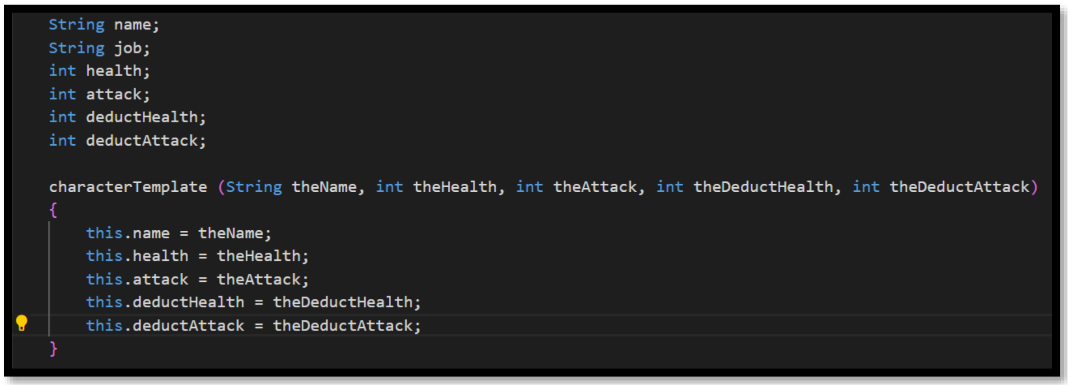
*Output after the change*

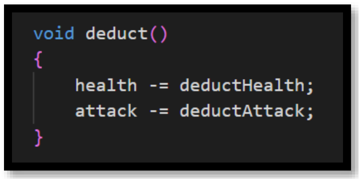

```
Cub Information:
Health: -88
Attack: 10
DEFEAT! You are dead!
BAD ENDING: DEAD
```

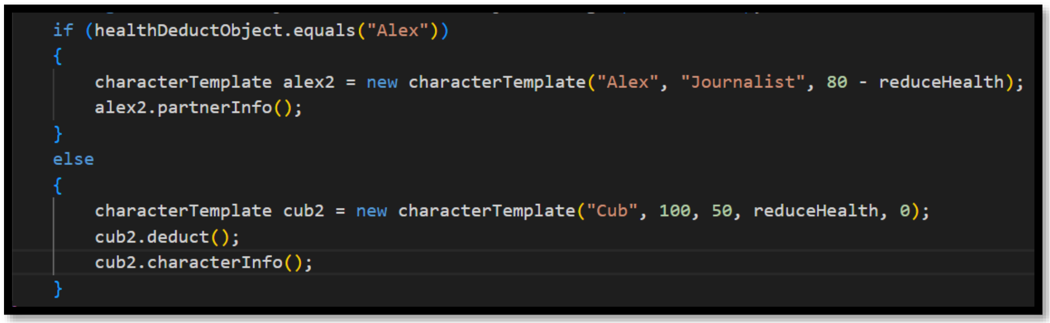
## 5.3 Efficiency Development

**Change the value in object directly instead of creating new object**

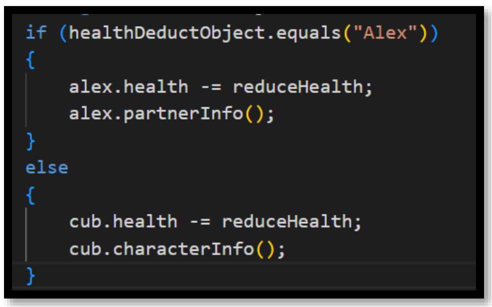
In order to reach the purpose of reduce health and add attack, I have created multiple objects for actually a same character. After the coding, it looks extremely redundant, which decrease the efficiency of the code implementation. Then, I have done some research, I realized that attribute variables could change directly in the main class using object's name + "." + variable's name.


```
String name;
String job;
int health;
int attack;
int deductHealth;
int deductAttack;

characterTemplate (String theName, int theHealth, int theAttack, int theDeductHealth, int theDeductAttack)
{
    this.name = theName;
    this.health = theHealth;
    this.attack = theAttack;
    this.deductHealth = theDeductHealth;
    this.deductAttack = theDeductAttack;
}
```


```
void deduct()
{
    health -= deductHealth;
    attack -= deductAttack;
}
```


```
if (healthDeductObject.equals("Alex"))
{
    characterTemplate alex2 = new characterTemplate("Alex", "Journalist", 80 - reduceHealth);
    alex2.partnerInfo();
}
else
{
    characterTemplate cub2 = new characterTemplate("Cub", 100, 50, reduceHealth, 0);
    cub2.deduct();
    cub2.characterInfo();
}
```

*Improved Code:*



```
if (healthDeductObject.equals("Alex"))
{
    alex.health -= reduceHealth;
    alex.partnerInfo();
}
else
{
    cub.health -= reduceHealth;
    cub.characterInfo();
}
```

Not only for this, there are multiple code designs explained in **Code explanations**, like 1.1, 3.1, 3.2, 4, and 6.2, are highly enhanced the efficiency of the program. Since these parts are already planned before coding, therefore they are not demonstrated in this section.

## 6. Words in the End

Programming is a process requires strong concentration, logic, creativity, and carefulness. I have applied all of the knowledge learned into this project, which is tough, but also meaningful that could show the application skill on coding. *Hoosntee Crisis* is the first game that I have made using java, but I believe it is not going to be the last one!