**PokemonGame User Manual**

A Java-based emulation of the popular trading card game, Pokemon.

Cachary Tolentino

CSCI-3327

**Table of Contents**

**Software Description3**

Detailed Description3.1

System Requirements3.2

**Installation Guide4**

**Class Overview7**

**Software Description**

**A Java-based emulation of the popular trading card game, Pokémon**.

**Detailed Description**

The PokemonGame program is an emulation of the popular trading card game, Pokémon. It uses multiple classes that represent the type of cards you might find in the actual trading card game, such as Charmander, a Fire Card, and even Bill the trainer card! Users will be able to play the game via an AI or by versing themselves. Users are also given the ability to choose the allocation of their deck, to ensure that everyone has almost full flexibility!

**System Requirements**

* A working device, primarily a desktop or laptop
* An IDE (ex: VSCode, Eclipse, etc…)
* Java JDK (Ver. 17 & up) & JRE (SE 17 & up)

**Installation Guide**

To start playing the game, you need to download a couple of files and place them into a folder. The files are as follows:

**A screenshot of a computer

AI-generated content may be incorrect.**

Once download, ensure that it is placed in a folder that you can access via your choice of Code Editor (such as VSCode). From there, simply open up your code editor, open the folder, find the folder name, and open it.

A screenshot of a computer

AI-generated content may be incorrect.

Lastly, you can start playing by simply running the “PlayPokemon.java” file. The terminal will output the following:

A screen shot of a computer

AI-generated content may be incorrect.

Happy playing! 😊

**Example Gameplay**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a black background

AI-generated content may be incorrect.**

**Ai beating me (multipler working)**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Ai retreated, bulb turned to pika**

**A screenshot of a computer program

AI-generated content may be incorrect.**

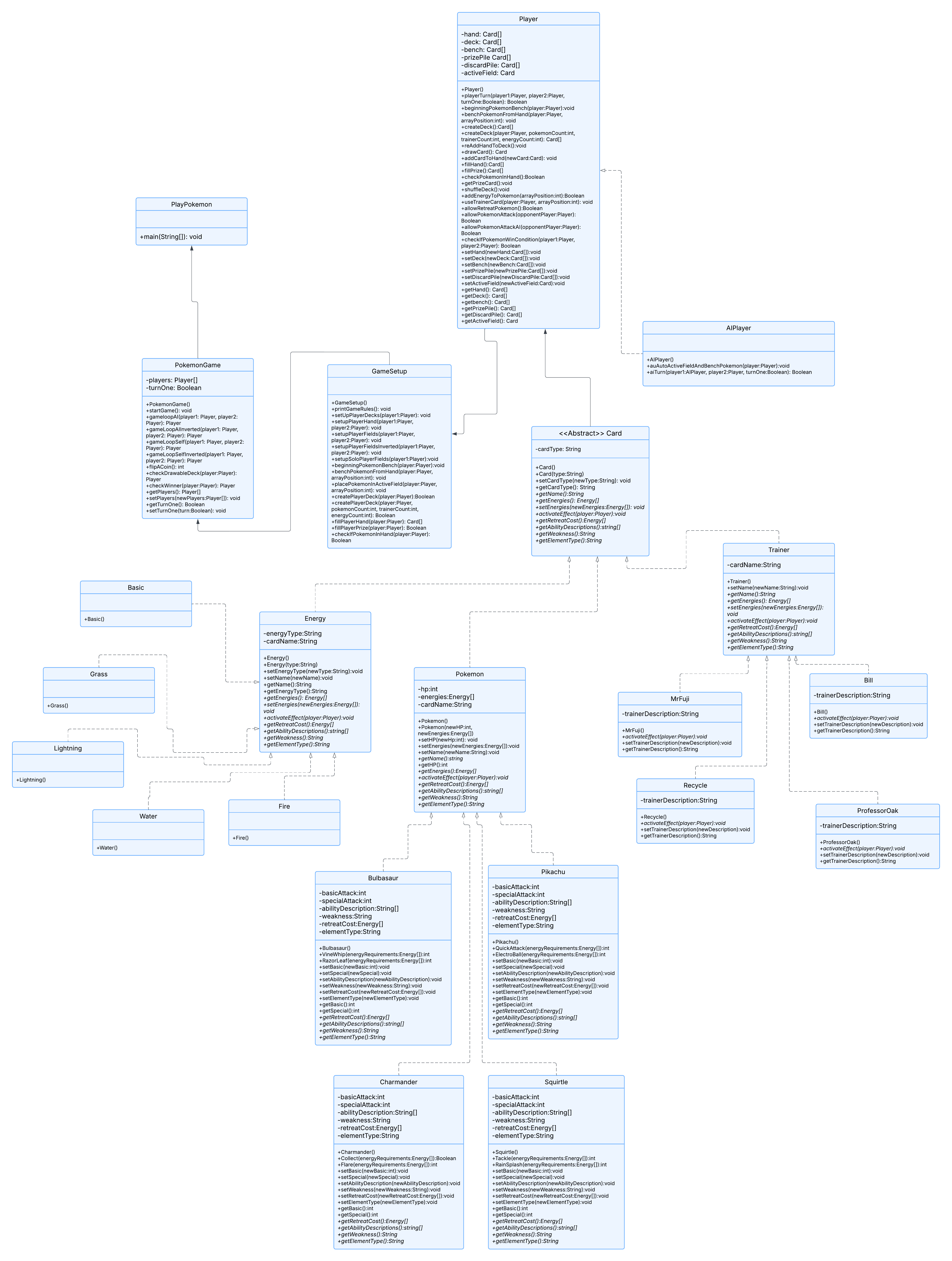
**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Class Overview**



The UML Diagram describes the relationship between all the classes within the program. PlayPokemon is simply the runner/tester class for running the entire program. This is associated with the PokemonGame class. The PokemonGame class is the main class that creates the entire game. Thus, the PlayPokemon class creates a PokemonGame object and calls the startGame() function.

The PokemonGame class is responsible for deciding the type of game loop that the user chooses (AI or Self) as well as checking each turn if and which player has won the game. This class is also associated with the GameSetup class. The GameSetup class is responsible for initiating the game’s initial objects such as the player objects, their fields such as their deck, active zone, hand, prize pile and discard pile.

The GameSetup class uses the Player class to create player objects. These player objects would represent each player and the information they would have. The Player class is responsible for performing any action that a player can perform. Some would be drawing a card, activating a trainer card’s effect, and attacking with their active Pokémon. Furthermore, Player has a specified child class, AIPlayer. The AIPlayer is specifically made for an AI game mode. This class is responsible for any actions that an AI can perform, which is primarily describing their turn action and automatically filling their fields.

Each player would have Cards. The Card class is an abstract class. It works as a framework for other types of Cards such as a Trainer card, Pokémon card, and Energy card. It still does contain a constructor to ensure that polymorphism would work when identifying the type of card it would be. It contains abstract methods that all other card objects would need to fulfill their role in the game scheme.

The types of Cards are mainly Energy, Trainer, and Pokémon. Thees classes are further subdivided with their child classes which defines specific types of their respective categories.

The Energy class has five children, Basic, Water, Fire, Grass, and Lightning. These classes simply serve the purpose of being an identifiable energy card that the Pokémon objects may use.

The Trainer class has four children: Bill, MrFuji, ProfessorOak, and Recycle. These classes have their own specific effects that a player can activate during their turn.

Lastly, the Pokémon class have four children classes: Squirtle, Bulbasaur, Pikachu, and Charmander. Each Pokémon class contains specific information that relates to that type of Pokémon. It primarily consist of information such as their HP, Basic Attack, Retreat Cost, and Weakness. Furthermore, each Pokémon has their own specific abilities that cost a certain amount of energy to perform. To retreat the Pokémon, there is also a required energy count.

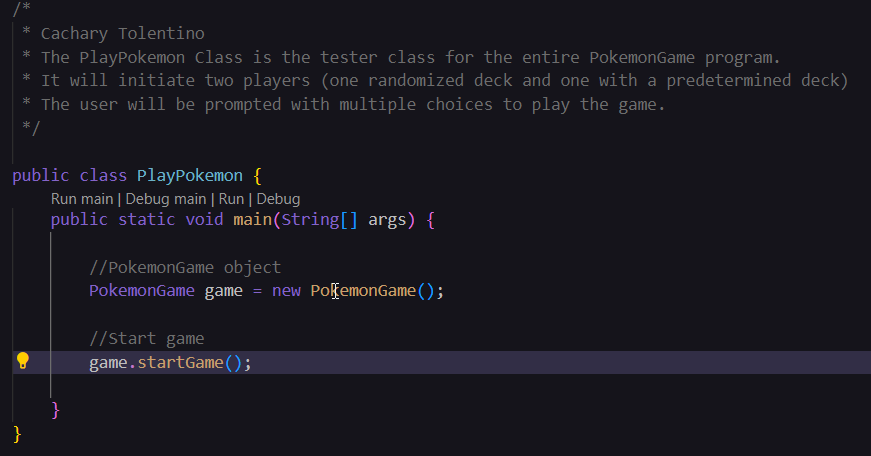
**Game Overview**

To play the game, two players are required. The user has the option of playing against an AI or against themselves. To do this, PokemonGame initializes two Player objects or one Player and one AIPlayer, depending on the game mode. Afterwards, the game will ask the user which side of the coin they would like to bet on. Depending on the result and the user’s guess, the first turn will belong to the one who guessed the correct side. Thus, inverting the player positions within the class. In the beginning, the player will have the option of customizing the allocation of their deck (i.e. the amount of trainer, Pokémon, and energy cards. It is important to note that the type of cards being made will be randomized despite custom allocation. In the AI game mode, the AI’s fields will be automatically filled depending on their drawn hand. For the player, they are given the choice of which Pokémon they want to start with and the chance to bench any extra Pokémon that they have on hand.

During the first round, the player with the first turn will have a restriction on which they cannot perform any attacks. But once the first turn is over, the restriction is lifted. In the game, the user has multiple options. They can attach an energy card onto their active Pokémon. This is restricted to once per turn. A player can also play an unlimited amount of trainer cards that they have in their hand. Furthermore, a player has a few options regarding their Pokémon’s. They have the option to bench any extra Pokémon from their hand. To note, the bench has a maximum of six Pokémon’s allowed. The player can also retreat their active Pokémon, although the Pokémon must have the required amount of energy to retreat attached to them. Lastly, they can make their Pokémon attack the opponent Pokémon. These attacks will only be successful given that Pokémon also has the required energy attached to them. If a player does not have any other option, then they can simply pass their turn.

**PlayPokemon Class**

The PlayPokemon class contains the main method. It creates a PokemonGame object and calls the startGame() function to initiate the game.

****

**PokemonGame Class**

The PokemonGame class contains two global variables: players and turnOne. The players variable holds the Player objects that would be made during the initialization of the game. These can be either both Player or one Player and one AIPlayer object. The turnOne variable is responsible for keeping track of whether the current turn is the first one or not. This is later changed right after a player/ai turn is over.

The class also contains an constructor that initializes each variable with their own values. Note: even though this initializes the players variable with Player objects, this one of the player objects will be changed to an AIPlayer object if the game mode selected is versus an AI.

**A screenshot of a computer program

AI-generated content may be incorrect.**

In the following code, we can see the initial layout of how the game would greet the player. This gives the player to cancel out of the game at the start and right before selecting the type of game mode. Here we can also see the mechanism for deciding who will be the first player. Once the player chooses a side of the coin, the opposite is automatically assigned to the other player.

A screenshot of a computer program

AI-generated content may be incorrect.

After a game mode is chosen, the following layout is repeated depending on whether the player chose to customize their deck or not. In the code, this shows a customized deck. For a non customized deck, the player will simply use the same method as the AI (createPlayerDeck). Here a new GameSetup object is made used to initialize all parts of the game for both the player and AI (or player and player). Lastly, the game loop will be called (depending on the results of the coin toss and game mode, this will vary). Once a winner is found, they will be announced and the program will terminate.

A computer screen shot of a game code

AI-generated content may be incorrect.

In the following code, we can see one of the four variants of the game loop. In total there is gameloopAI, gameLoopAIInverted, gameLoopSelf, and gameLoopSelfInverted. The difference between the inverted and non-inverted versions is simply swapping the chunks of code for either the player or AI (or player) in which they may have the first turn. Here we can also see that the turnOne variable will be changed to false immediately after a turn is performed. Any of the turn functions will return a player if that player is found to be a winner or that the current player has no more cards in their deck, thus losing the game. The checkWinner function simply checks if the current player has an empty prize pile, thus defeating six Pokémon which makes them the winner.

A screenshot of a computer program

AI-generated content may be incorrect.

The following code shows some functions that is used by the class primarily for some of the tasks such as flipping a coin to decide which player will be first, checking if a player still has cards in their deck, and if the player has obtained all their prize cards, thus winning the game.

A screenshot of a computer program

AI-generated content may be incorrect.

In the code below, we can see the setters and getters for the global variable of the class.

A screenshot of a computer program

AI-generated content may be incorrect.

**GameSetup Class**

The GameSetup class contains no global variables. Since it mainly functions as a helper class class for the PokemonGame class it does not need any. Therefore, its constructor does not initialize any variables.

**A screenshot of a computer program

AI-generated content may be incorrect.**

The printGameRules function simply prints all the rules, game setup, a how to play guide, and the win conditions for the game.

**A screenshot of a computer program

AI-generated content may be incorrect.**

The shown function will be used whenever a player has decided to create a custom allocated deck in the start of the game. It simply asks for three inputs and uses those inputs as arguments for creating the deck of the player.

**A screen shot of a computer program

AI-generated content may be incorrect.**

In the following code, the function is used right after each player’s deck has been created. These draws a hand for the players automatically. It also implements a mechanism which prevents any mulligan hand to appear, it will simply reshuffle the hand into the deck and redraw until atleast one Pokémon is in the current hand.

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of text

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Player Class**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A computer screen with text

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of text

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**AIPlayer Class**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Abstract Card Class**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Energy(Basic, Grass, Water, Fire, Lightning) Class**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Pokemon(Bulbasaur, Pikachu, Charmander, Squirtle) Class**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of text

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Trainer(MrFuji, ProfessorOak, Recycle, Bill) Class**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**