



eduresourceza.com

Downloaded from eduresourceza.com
Pokémon day-care needs your help!!

SECTION A

OOP

The owners of the Pokémon day-care have suddenly had a mass amount of eggs hatch in their possession and they have been overwhelmed by the amount of Pokémon to process. All the Pokémon that have hatched have been saved in a computer file **Pokemon.txt** in the format:

{Name}#{type}#{amount of stages}#{gender:m/f}

```
Bulbasaur#grass#3#m
Charmander#fire#3#m
Squirtle#water#3#f
Ponyta#fire#2#f
Lapras#other#1#m
Ekans#other#2#f
Pinsir#other#1#m
Vulpix#fire#2#f
Magikarp#water#2#f
Eggsecute#grass#2#m
Rattata#other#2#m
Pichu#other#3#m
Shelder#water#2#m
Mankey#other#2#f
Staryu#water#2#f
Mewtwo#other#1#m
Dratini#other#3#m
Growlithe#fire#2#m
Tentacool#water#2#f
Bellsprout#grass#3#f
```

Question 1

- 1.1 Create a class called **Pokemon** to store the following data for each Pokemon: name, type, amount of stages, gender, evasion, accuracy. (3)
- 1.2 Create a constructor to assign values for name, type, amount of stages and gender. (3)
- 1.3 In the constructor randomly generate values for **evasion** and **accuracy**. The random numbers must lie between 50 and 150 (incl). (4)
- 1.4 Code accessor and mutator methods for each field. (4)
- 1.5 Code a **toString** method to display all the fields separated by tabs. (4)

[18]

Question 2

The day-care couple have also stated that they will **never have more than 150 Pokémon at once** and that it would be unnecessary to make arrangements for more entries than that.

- 2.1 Create a class called **PokemonArray** with fields to store 150 Pokemons and a field to remember the number of Pokemons stored. (3)
- 2.2 Code a constructor method to instantiate an array of Pokemon objects using the array you created in 2.1 and the text file. (10)
- 2.3 Code a **toString** method to display each Pokemon one after the other. (4)
- 2.4 Code a method to **search** for a Pokemon's details given their position in the array. The method must return a Pokemon object. (6)
- 2.5 Code a method that will return the number of elements in the array. Name this method **getSize**. (3)

- 2.6 Code a method called **getMove** that will randomly generate one of the following moves: Dragon Rush, Fly, Jump Kick, Mud Bomb, Outrage and Power Whip. Use a constant array to help you. Return the name of the move as a String. (6)
- 2.7 Code typed method called **doBattle** that will accept 2 integers as parameters and return a String stating who won the battle. The two integer parameters will be two Pokemon that will fight. Use the search method you created in 2.5 to search for the details of each Pokemon. (6)
- 2.7.1 Add code to call the **getMove** method for each Pokemon and get the name of the move. Make sure that both Pokemon's have a different move. (5)
- 2.7.2 Each move has a different Move Accuracy: Dragon Rush is 90, Fly 100, Jump Kick 85, Mud Bomb 99, Outrage 73 and Power Whip 96. Assign an accuracy value to each move, coding your solution as efficiently as possible. (5)
- 2.7.3 The result of a battle can be predicted with the formula:
- $$Prediction = Move Accuracy * \frac{Pokemon Accuracy}{Pokemon Evasion}$$
- Calculate the prediction for each Pokemon's move. (3)
- 2.7.4 The Pokemon with the higher prediction wins the move. Return a message from the doBattle method stating the name of the Pokemon that won. (5)

[56]

Question 3

- 3.1 Code an application called **PokemonApp** to instantiate a **PokemonArray** object. (2)
- 3.2 Add code to display all the Pokemon. (2)
- 3.3 Add code to ask the user to input 2 integers between 0 and the size of the array (using the **getSize** method) and call the **doBattle** method of the PokemonArray class. Display the name of the winning Pokemon on the screen. You may assume that the user enters the correct values. (2)

[6]

SECTION B

SQL

The database Pokemon has two tables one to store details about the Characters and one to store the Moves of each Pokemon.

Character Table in Design View

Characters	
Field Name	Data Type
No	Number
Name	Text
Type	Text
Hit Points	Number
Attack	Number
Defence	Number
Special Attack	Number
Special Defence	Number
Speed	Number
Total	Number

Sample Data of the **Characters** table

No	Name	Type	HitPoints	Attack	Defence	SpecialAtta	SpecialDefe	Speed	Total
634	Zweilous	Dark	72	85	70	65	70	58	420
570	Zorua	Dark	40	65	40	80	40	65	330
571	Zoroark	Dark	60	105	60	120	60	105	510
263	Zigzagoon	Normal	38	30	41	30	41	60	240
523	Zebstrika	Electric	75	100	63	80	63	116	497
145	Zapdos	Electric	90	90	85	125	90	100	580
335	Zangoose	Normal	73	115	60	60	60	90	458

Moves table in **Design** view

Field Name	Data Type
Name	Text
Type	Text
Category	Text
Power	Number
Accracy	Number
Power Points	Number
Effect	Text

Sample Data of the **Moves** table

Name	Type	Category	Power	Accracy	Power Point	Effect
Absorb	Grass	Special	20	100	25	User recovers l
Acid	Poison	Special	40	100	30	May lower opp
Acid Armor	Poison	Status			40	Sharply raises
Acid Spray	Poison	Special	40	100	20	Sharply lowers
Acrobatics	Flying	Physical	55	100	15	Stronger when
Acupressure	Normal	Status			30	Sharply raises
Aerial Ace	Flying	Physical	60		20	Ignores Accura
Aeroblast	Flying	Special	100	95	5	High critical hit
After You	Normal	Status			15	Gives target pr
Agility	Psychic	Status			30	Sharply raises
Air Cutter	Flying	Special	55	95	25	High critical hit
Air Slash	Flying	Special	75	95	20	May cause flin

Open the document **Answers** and paste your answers to the following questions.

- Sort the Moves table by category and then by type. (3)
- List the top 10 moves whose difference between power and their power points is the greatest. (4)
- Check if the data has been entered correctly for each Pokemon Character by calculating the total of hit points, attack, defence, special attack, special defence and speed. Display the records whose fields do not add up to the Total field. (5)
- Count the number of Pokemons of each type. (5)
- Add a new Pokemon using your name and any type. Randomly generate a value for speed between 80 and 120. (5)
- Change all Special category Moves to be Swimming category. (4)
- Find any Move that has the word "raises" in its Effect. List the Name and Effect. (4)
- Create a code for each Pokemon using the first three letters of their Name and double the Hit Points. (5)
- Find the average Power Points of each type of Move. (5)