



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. A Pokemon can evolve to a different Pokemon and their previous name must be stored.

All the Pokémon that have hatched have been saved in a computer file **Pokemon.txt** in the format:

Pokemon.txt

Name#Type#Amount of stages#Gender

```
Bulbasaur#Grass#3#M
Charmander#Fire#3#M
Squirtle#Water#3#F
Ponyta#Fire#2#F
Lapras#Flying#1#M
Ekans#Normal#2#F
Pinsir#Ghost#1#M
Vulpix#Fire#2#F
Magikarp#Water#2#F
Eggsecute#Grass#2#M
Rattata#Bug#2#M
Pichu#Poison#3#M
Shelder#Water#2#M
Mankey#Psychic#2#F
Staryu#Water#2#F
Mewtwo#Ice#1#M
Dratini#Steel#3#M
Growlithe#Fire#2#M
Tentacool#Water#2#F
Bellsprout#Grass#3#F
```

A Pokemon can have different Moves. A Pokemon can use a move that is of the same type as the Pokemon. For example Bulbasaur can only perform moves of type Grass.

A sample of the text file **Moves.txt** is shown below:

Name#Type#Category#Power#Accuracy#Power Points#Effect.

```
Absorb#Grass#Special#20#100#25#User recovers half of the HP inflicted on
opponent.
Acid#Poison#Special#40#100#30#May lower opponent's Special Defense.
Acid Armor#Poison#Status#-#-#40#Sharply raises user's Defense.
Acid Spray#Poison#Special#40#100#20#Sharply lowers opponent's Special Defense.
Acrobatics#Flying#Physical#55#100#15#Stronger when the user does not have a held
item.
Acupressure#Normal#Status#-#-#30#Sharply raises a random stat.
Aerial Ace#Flying#Physical#60##20#Ignores Accuracy and Evasiveness.
```

Two Pokemon's can do battle each using a move and by comparing their power points multiplied by a luck factor between 1 and 5. The Pokemon with the highest result wins.

Question 1

- 1.1 Create a class called **Pokemon** to store the following data for each Pokemon: name, type, amount of stages and gender. (3)
- 1.2 Create a parameterised constructor to assign values for name, type, amount of stages and gender. (3)
- 1.4 Code accessor methods for the name and type field. (3)

Downloaded from eduresourceza.com

- 1.5 Code a toString method to display all the fields separated by tabs. (4)

[13]

Question 2

- 2.1 Create a class called **Evolved** class that inherits from the Pokemon class and to store the previous name of a Pokemon that has evolved. (3)
- 2.2 Code a parameterised constructor to accept the previous name of the Pokemon to assign a value to the previous name field and to call the constructor of the super class to assign values to the super class fields. These values must also be passed as a parameter. The Pokemon super class must store the current name. (3)
- 2.3 Code a **toString** method to display the details of the Pokemon including the previous name field using the super class. (3)

[9]

Question 3

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.

- 3.1 Create a class called **PokemonArray** with fields to store 150 Pokemon and a field to remember the number of Pokemon stored. (2)
- 3.2 Code a constructor method to instantiate an array of Pokemon objects using the array you created in 3.1 and the text file **Pokemon.txt**. (8)
- 3.3 Code a **toString** method to display each Pokemon one after the other. (4)
- 3.4 Code a method called **evolve** to change Ekans to Arbok and Bulbasaur to Ivysaur. Both Pokemons must be instantiated as an **Evolved** class, storing their new name as well as their previous name. (6)
- 3.5 Code a method called **chooseMove** that will return the name and power of the move given the type of Pokemon as a parameter. The method must scan through the text file **Moves.txt** and find the first type of move that matches the Pokemon type. For example Ponyta has a type of Fire. The first move of type Fire in the text file **Moves.txt** is

Blast Burn#Fire#Special#150#90#5#User must recharge next turn.
The method must return Blast Burn and the power 5 combined with a # i.e. Blast Burn#5.

Note that some fields do not have a value and is shown as a '-'. Your program must cope with these values. (8)

- 3.6 Code typed method called **doBattle** that will randomly generate two integer positions in the array. Use these positions to identify the 2 Pokemons who will take part in a battle. (4)
- 3.6.1 Ensure that the two positions are not the same. (3)
- 3.6.2 Call the **chooseMove** method for each Pokemon to extract the power points for each Pokemon's move. (2)
- 3.6.3 Multiply each Pokemon's power points by a random factor of 1 to 5. (4)
- 3.6.4 Determine the winning Pokemon with the most power points multiplied by the factor. (3)

- 3.6.5 Create a message that is returned to the calling application that will return the names of the Pokemons that fought, the result of their power points multiplied by the factor and who won the battle.

For example if Ponyta fights Pinsir and a 3 is generated for Ponyta and a 4 is generated for Pinsir the built up message would be:

```
Ponyta fought Pinsir
Ponyta power points is 5 times 3 = 25
Pinsir power points is 10 times 4 = 40
Pinsir wins.
```

(6)

[50]

Question 4

- 4.1 Code an application called **PokemonApp** to instantiate a **PokemonArray** object. (2)
- 4.2 Add code to display all the Pokemons. (2)
- 4.3 Add code to call the **evolve** method and then display all the Pokemon as well as the Evolved objects. (2)
- 4.4 Add code to call the **doBattle** method and display the resulting text message on the screen. (2)

[8]

SECTION A TOTAL

[80]

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	SpecialAttac	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	Zigagoon	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 characters alphabetically who have a total points value over 300. (3)
- List all the Moves that have no values for Power. (3)
- Calculate the HitPoints average of all the Dark type Pokemons. Name the average AVE. (5)
- Find the Moves with the highest PowerPoints in each category. (5)
- Add a new Move using any name and any type. (3)
- Reduce all the Pokemons Speed values by 5 points. (3)
- List all the Pokemons with their available moves sorted in descending order of points. (5)
- Count the number of Pokemons whose move category is Special or Physical. (7)
- List all the Pokemons names with their moves as well as the effect of the move. Show only the effects that will hit more than once or have a high critical hit ratio. (6)

SECTION B TOTAL

[40]

TOTAL

[120]

eduresourceza.com