



# JAVA PROJECT

OBJECT ORIENTED PROGRAMMING IN JAVA

---



BEE29 Aishwarya Kamble  
BEE13 Durgesh Kolte  
BEE10 Yash Pamnani



# Contents

- 1) Software used
- 2) Java concepts implemented
- 3) Pet Shop Application interface
- 4) MySQL Database interface

# SOFTWARE USED

## 1.) Netbeans IDE: -

- Written in Java
- Cross-Platform Support
- Fast And Smart Editing
- GUI Builder
- Easy and Efficient Project Management
- Smart Error Analyzer



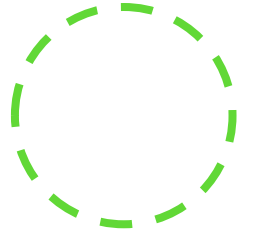
## 2.) MySQL: -

- Written in C, C++
- Open Source Platform
- Supports many languages like PHP, C, C++, JAVA, etc.
- Quick and works well even with large data sets
- Table is customizable using simple commands
- User-friendly, easy to link MySQL databases with different application


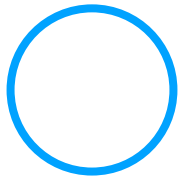




# JAVA CONCEPT IMPLEMENTED




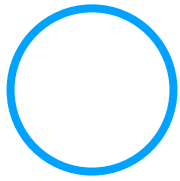
**Object-Oriented Programming** is a methodology or paradigm to design a program using classes and objects.

- 1. Classes**- *Collection of objects* is called class. It is a logical entity. A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.
  - 2. Objects**- An Object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicate without knowing the details of each other's data or code.
  - 3. Inheritance**- *When one object acquires all the properties and behaviors of a parent object*, it is known as inheritance. It provides code reusability.
- 
- 

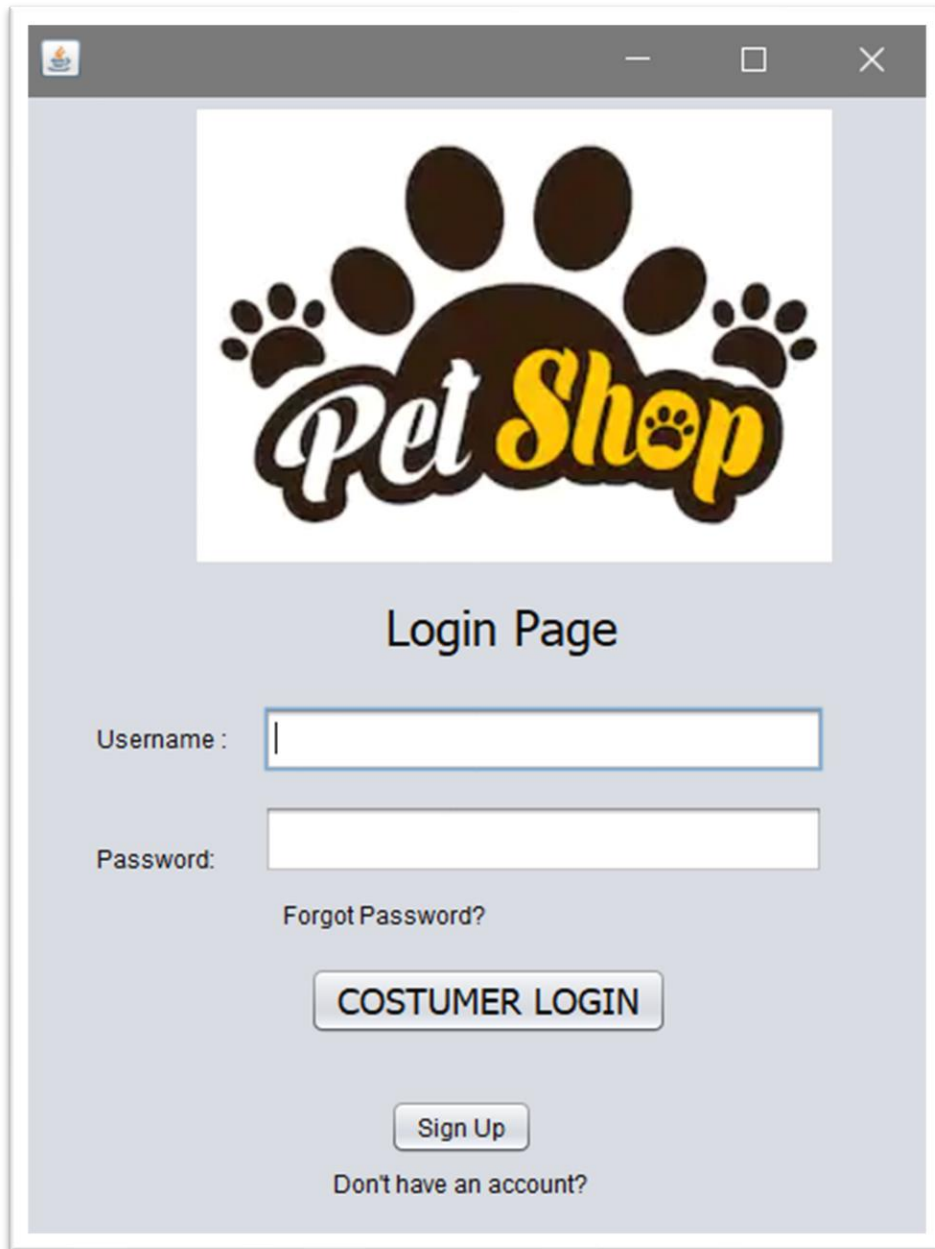


# JAVA CONCEPT IMPLEMENTED




- 4. **Constructor**- constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.
  - 5. **Package**- package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.
  - 6. **Exception Handling**- Exception Handling is one of the powerful *mechanism to handle the runtime errors* so that normal flow of the application can be maintained.
- 
- 

# PET SHOP APPLICATION



The login page features a header with a small icon and window controls. Below this is a large logo for 'Pet Shop' with a paw print design. The main content area is titled 'Login Page' and contains two input fields for 'Username' and 'Password'. A 'Forgot Password?' link is positioned below the password field. At the bottom, there is a 'COSTUMER LOGIN' button and a 'Sign Up' button, with the text 'Don't have an account?' centered below the 'Sign Up' button.



## Login Page

Username :

Password:

[Forgot Password?](#)

Don't have an account?



The sign up form has a header with a small icon and window controls. The title 'SignUp Form' is centered at the top. The form includes four input fields: 'Name', 'Contact No.', 'Address', and 'Username'. Below the 'Username' field is a 'Password' field. A 'Log In' button is located at the bottom right of the form.

## SignUp Form

Name:

Contact No.:

Address:

Username:

Password:

# PET SHOP APPLICATION



I'm not a robot





  
reCAPTCHA

652

Match the given number:

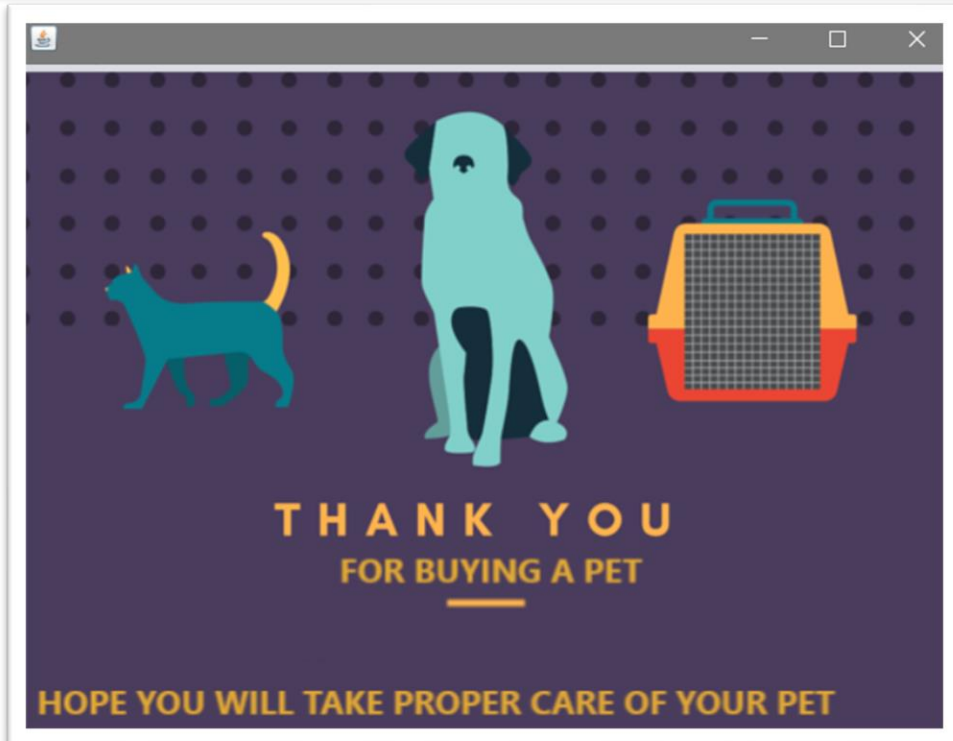
Username:

Select your Pet



# PET SHOP APPLICATION

id	type	breed	sex	quantity	price
8	Dog	Labrador	M	1	10000
9	Dog	Beagle	F	1	23000
10	Dog	German Shep...	F	1	7000
11	Dog	Dobermann	M	1	14000
12	Dog	Pug	F	1	9000
13	Dog	Pomeranian	F	1	5000
14	Dog	Golden Retrie...	M	1	20000



Enter the Pet you need to buy

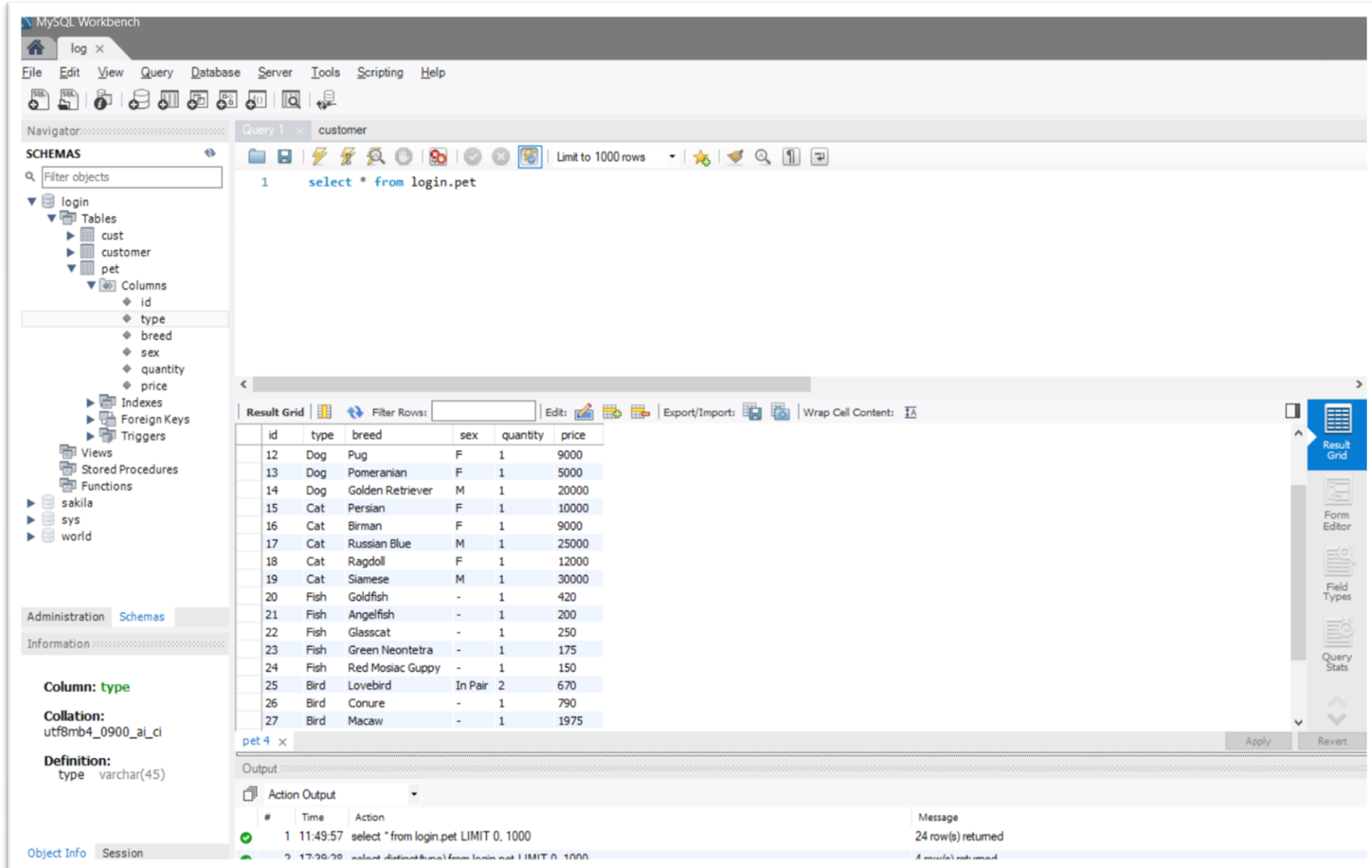
Pet Type :

Breed :

Quantity :



# MYSQL DATABASE- PET DATABASE



The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view of the database structure, including tables like 'login', 'customer', and 'pet'. The 'pet' table is selected, showing its columns: id, type, breed, sex, quantity, and price. Below the schema tree, the 'Column: type' is highlighted, showing its collation as 'utf8mb4\_0900\_ai\_ci' and its definition as 'type varchar(45)'. The main query editor shows a query: `select * from login.pet`. The 'Result Grid' pane displays the query results as a table with 24 rows. The 'Output' pane at the bottom shows the execution log, indicating that the query was executed successfully and returned 24 rows.

id	type	breed	sex	quantity	price
12	Dog	Pug	F	1	9000
13	Dog	Pomeranian	F	1	5000
14	Dog	Golden Retriever	M	1	20000
15	Cat	Persian	F	1	10000
16	Cat	Birman	F	1	9000
17	Cat	Russian Blue	M	1	25000
18	Cat	Ragdoll	F	1	12000
19	Cat	Siamese	M	1	30000
20	Fish	Goldfish	-	1	420
21	Fish	Angelfish	-	1	200
22	Fish	Glasscat	-	1	250
23	Fish	Green Neontetra	-	1	175
24	Fish	Red Mosaic Guppy	-	1	150
25	Bird	Lovebird	In Pair	2	670
26	Bird	Conure	-	1	790
27	Bird	Macaw	-	1	1975

Column: type  
Collation: utf8mb4\_0900\_ai\_ci  
Definition: type varchar(45)

Query 1: `select * from login.pet`

Result Grid:

id	type	breed	sex	quantity	price
12	Dog	Pug	F	1	9000
13	Dog	Pomeranian	F	1	5000
14	Dog	Golden Retriever	M	1	20000
15	Cat	Persian	F	1	10000
16	Cat	Birman	F	1	9000
17	Cat	Russian Blue	M	1	25000
18	Cat	Ragdoll	F	1	12000
19	Cat	Siamese	M	1	30000
20	Fish	Goldfish	-	1	420
21	Fish	Angelfish	-	1	200
22	Fish	Glasscat	-	1	250
23	Fish	Green Neontetra	-	1	175
24	Fish	Red Mosaic Guppy	-	1	150
25	Bird	Lovebird	In Pair	2	670
26	Bird	Conure	-	1	790
27	Bird	Macaw	-	1	1975

Output:

#	Time	Action	Message
1	11:49:57	select * from login.pet LIMIT 0, 1000	24 row(s) returned
2	17:30:20	select distinct(breed) from login.pet LIMIT 0, 1000	4 row(s) returned

# MYSQL DATABASE- CUSTOMER DATABASE

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with a search filter. The 'login' schema is expanded, showing tables 'cust', 'customer', and 'pet', along with their columns, indexes, foreign keys, triggers, views, stored procedures, and functions. The 'customer' table is selected. The main query editor shows a query: `select * from login.cust`. The 'Result Grid' tab is active, displaying the query results in a table format. The 'Output' tab is also active, showing the execution log with timestamps and messages for each statement executed.

**Table: cust**

**Columns:**

- idcust: int AI PK
- cName: varchar(45)
- contact: varchar(45)
- email: varchar(45)
- pass: varchar(45)
- address: varchar(45)

**Result Grid**

	idcust	cName	contact	email	pass	address
▶	8	yash	7452136484	yashp	yashp	Ahmednagar
▶	9	durgesh	7853641875	dk	abc	Mumbai
▶	10	aishwarya	9563241785	aishwarya	pqr	Pune
*	NULL	NULL	NULL	NULL	NULL	NULL

**Output**

#	Time	Action	Message
✓	23 03:07:54	DELETE FROM login.cust WHERE idcust = 5	1 row(s) affected
✓	24 03:07:57	DELETE FROM login.cust WHERE idcust = 6	1 row(s) affected
✓	25 03:08:02	DELETE FROM login.cust WHERE idcust = 7	1 row(s) affected
✓	26 03:08:07	DELETE FROM login.cust WHERE idcust = 8	0 row(s) affected
✓	27 03:10:40	select * from login.cust LIMIT 0, 1000	3 row(s) returned
✓	28 03:11:40	select * from login.cust LIMIT 0, 1000	3 row(s) returned



**Thank You 😊**

**Any Questions ??**