

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**PET REGISTRATION**

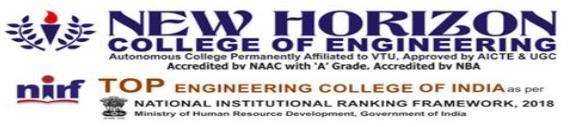
**A MINI PROJECT REPORT**

***Submitted by* RANJITHA AR USN: 1NH18CS157**

***In partial fulfilment for the award of the degree of***

### BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

*Certificate*

*This is to certify that the mini project work titled PET REGISTRATION*

*Submitted in partial fulfilment of the degree of Bachelor of Engineering*

USN: 1NH18CS157

*DURING*

*EVEN SEMESTER 2019-2020 COURSE CODE: 19CSE39*

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

*Name of the Examiner Signature with date*

1.

2.

**ABSTRACT**

The main idea of this program is to provide an Online portal for pet shop using Graphical User Interface so that the customers who are willing to register their dog’s details or file a missing compliant in the website. This program will display the different criteria’s in which customers can login or register or file a missing complaint. If the customer is already registered in the portal then then can directly login in through their username and password.

If the customer doesn’t have a account then he must register in the registration form and fill the necessary details and then they will also get registered. If the customer wants to file a missing compliants then he/she must click on the missing compliants box and they should

fill the necessary details and even they should attach a photo of their dog which will be easier to find the dog. And there is also a box called NGO’S Rescued pets where they find details of some dogs names which are been rescued by the NGO’S. We hve also used events, text fields, button and frame. Different inbuilt methods such as setVisible(),add(),setBounds(),setSize(),dispose() and setLayout().

A online portal pet shop Is an element that focuses on the various needs of the customers and theirdogs. Even in this portal we can register the dogs details if we don’t have a account earlier, if we have account earlier then we can just logon in to the portal through our respected username and passwordand even we can file compliant on the missing dogs and even we can see the rescued dogs by theNGOS.

Problem Definition :

The main reason to develop this type of program is to make the customers feel comfortable for all

their needs like registration of their dogs details or even they can file missing compliant and even

they can even see the dogs rescued by the NGOS.

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr.Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment

I take this opportunity to express my profound gratitude to **Dr.Manjunatha,** Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr.Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank Dr**. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Ms.Asha Borah,** assistant professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

#### RANJITHA AR

**(1NH18CS157)**

|  |  |  |
| --- | --- | --- |
| **CONTENTS** |  | |
| ABSTRACT ACKNOWLEDGEMENT  LIST OF FIGURES | III  VII | IV |
| **1. INTRODUCTION** |  |  |
| 1.1. PROBLEM DEFINITION |  | 1 |
| 1.2. OBJECTIVES |  | 1 |
| 1.3. METHODOLOGY TO BE FOLLOWED |  | 1 |
| * 1. EXPECTED OUTCOMES   2. REQUIREMENTS   1.51 HARDWARE REQUIREMENTS | 1 | 1 |
| 1.52 SOFTWARE REQUIREMENTS | 1 |  |
| **2. OBJECT ORIENTED CONCEPTS** |  | 2 |
| * 1. CLASS      1. TYPES OF VARIABLES IN CLASS | 2 | 3 |
| 2.2 OBJECT  2.3. INHERITENCE | 8 | 5 |
| 2.3.1IMPORTANT TECHNOLOGIES OF INHERITANCE | 9 |  |
| 2.3.2 DIFFERENT FORMS OF INHERITANCE | 10 |  |

* 1. POLYMORPHISM 12
     1. TYPES OF POLYMORPHISM
  2. [ABSTRACT CLASS 12](#_TOC_250001)
  3. [MULTITHREADING 13](#_TOC_250000)
     1. BENEFITS OF MULTITHREADS
     2. DIFFERENT STAGES OF LIFE CYCLE OF MULTITHREADS

|  |  |  |
| --- | --- | --- |
| 2.7 I/O FUNCTIONS | 14 |  |
| 2.8 JAVA PACKAGES | 15 |
| 2.9 EXCEPTION HANDLING | 16 |
| **3. DESIGN** |  |
| 3.1. ALGORITHM |  | 23 |
| **4. IMPLEMENATION** |  | 26 |

1. **RESULTS** 27
2. **CONCLUSION** 33
3. **REFERENCE** 34

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. No** | **Figure Description** | **Page No** |
| 5.1 | RESULTS STEP 1 | 27 |
| 5.2 | RESULTS STEP 2 | 28 |
| 5.3 | RESULTS STEP 3 | 28 |
| 5.4 | RESULTS STEP 4 | 29 |
| 5.5 | RESULTS STEP 5 | 30 |
| 5.6 | RESULTS STEP 6 | 31 |
| 5.7 | RESULTS STEP 7 | 32 |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER 1**

**INTRODUCTION**

An online portal pet shop is an element that focuses on the various needs of the customers and their dogs. Even in this portal we can register the dogs details if we don’t have a account earlier, if we have account earlier then we can just logon in to the portal through our respected username and password and even we can file compliant on the missing dogs and even we can see the rescued dogs by the NGOS.

### Problem Definition:

The main reason to develop this type of program is to make the customers feel comfortable for all their needs like registration of their dogs details or even they can file missing compliant and even they can even see the dogs rescued by the NGOS.

### HARDWARE REQUIREMENTS

* + - Processor : Any of the Processor above 500 MHz
    - RAM : 512Mb
    - Hard Disk : 10 GB
    - Input device : Standard Keyboard , Mouse
    - Output device : VGA , High Resolution Monitor

### SOFTWARE REQUIREMENTS

* Operating system : Windows XP
* Front End : ASP.Net 2.0
* IDE : Visual Studio 2008
* Data Base : SQL Server Management Studio 2005
* Server : Internet Information Services
* Database Connectivity : ODBC Sources (with SQL Server)

**CHAPTER 2**

**OBJECTS ORIENTED CONCEPTS**

### Classes in Java:

* + - class is like a blueprint or a template for creating objects in java. It defines the state or behavior of the object created. class can have any number of variables, and methods of various types to access to different values.,
    - Each class has a constructor, it can be of type default or parameterized. These constructors are used to initialize objects, with default values. class can also inherit characteristics from other class.
    - While defining a class, we can declare its exact form and nature, by specifies the data that it contains and the code which operates on the data.
    - The class is declared by use of the class keyword. The general form of a class definition is as follows:

Class classname

{

Type instance-variable1; Type instance-variable2;

//…..

Type instance variable N;

Type methodname1 (parameter-list) {

//body of the method…

}

Type methodname2 (parameter-list) {

//body of the method…-

//………..

Type methodnameN (parameter-name){

//body of the method…

}

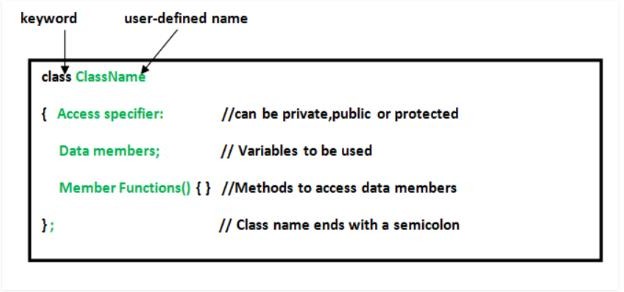
* + - The data/variables, defined within a class are called instance variables. The code is contained within methods. Collectively, the methods and the variables define within a class are called members of the class. In most classes, the instance variables are acted upon and accessed by the methods defined for that class. Thus, the methods determine how a class’ data can be used.
    - Putting the member elements and methods into together in the definition of a class is called encapsulation
    - A class declaration can include these in order:

Access Modifier -> Class name -> Superclass -> Interface -> Body There are various types of classes such as:

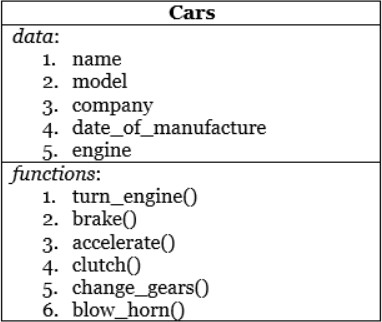
* Nested class
* Anonymous class
* Lambda expressions

#### Syntax:

Class ClassName



#### Fig 2.1 (a)



**Fig 2.1 (b)**

* + - The data,or variables,defined wothin a class are called instance variables.The code contained within methods.Collectively the methods and variableddefined within a class are called members of the class. In most classes,the instance variables are in acted upon and accessed by the methods defined that class.thus the methods determine how a class data can also be used.
    - Putting the number elements and methods into together in the definition of a class is called encapsulation.

### STACK CLASS:

* + - * Stacks are controlled as two ways called push and pop.
      * To put an item on top of the stack, we will use push.
      * To take an item off the stack, we will use pop.
      * Here is a class called stack which implements a stack for integers.

### OBJECTS:

* + - Objects have states and behaviors. Example: A dog has a states color, name, breed as well as behaviors -wagging,and barking and eating. An object is an instance of a class.
    - Obtaining objects of a class is two-step process.
      * first, must declare a variable of the class type
      * second, must acquire an actual, physical copy of the object and assign it to that variable, using the new operator.
    - The new operator dynamically (at run time) allocates memory for an object and returns a reference to it. This reference is the address in the memory of the object allocated by new.

**Syntax:** Box mybox = new Box(); class-var = new classname( );

* + - class-var is a variable of the class type is created. The classname is the name of the class that is being instantiated. The class name is followed by parentheses specifies the constructor for the class.
    - It can be rewritten to show each step more clearly which includes, reference to object and allocate a Box object:

Box mybox;

mybox = new Box ();

#### Assigning Object Reference Variables

Box b1 = new Box (); Box b2 = b1;

b1

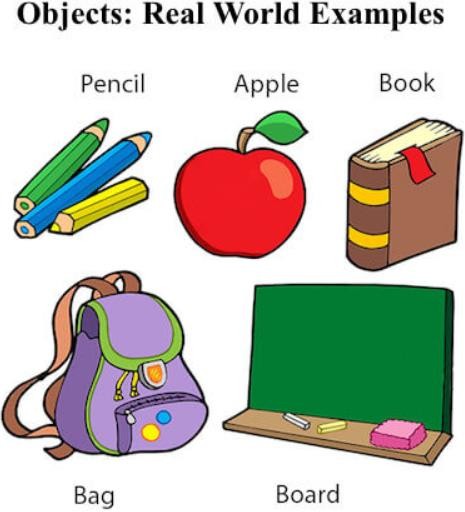
|  |  |  |
| --- | --- | --- |
|  | width |  |
|  | | |
| height | | |
|  | | |
|  | depth |  |

box object

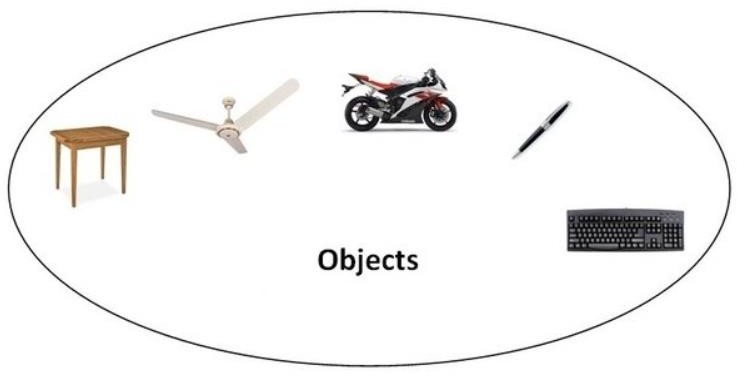
b2

2.2 (a)

* + - The assignment of b1 to b2 did not allocate any memory. It simply a makes b2 refer to the same object as does b1. Thus, that any changes made to the object through b2 will affect the object to which is referring to b1, since they are the same object.



#### Fig 2.2 (b)



**Fig 2.2 (c)**

**INTRODUCING METHODS:**

* + - Classes usually consist of two types: instance variables and methods. This is the general form of a method:

type name (parameter-list)

{

// body of method

}

* + - Here, type specifies the type of data returned by the method. This can be any valid type, including class types that we create. If method doesn’t return a value, then its return type must be void. The name of the method is specified by name. The parameter-list is a sequence of type and identifier pairs separated by commas.

#### Example: A program that adds void method to the Box class

class Box

{

double width; double height; double depth; void volume ()

{

System.out.print ("Volume is "); System.out.println (width \* height \* depth);

}

}

class BoxDemo3

{

public static void main(String args[])

{

Box mybox1 = new Box(); Box mybox2 = new Box(); mybox1.width = 10;

mybox1.height = 20;

mybox1.depth = 15;

mybox2.width = 3;

mybox2.height = 6;

mybox2.depth = 9; mybox1.volume (); mybox2.volume ();

}

}

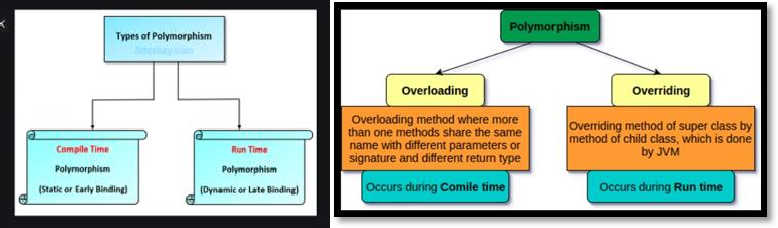
#### Output:

Volume is 3000.0

Volume is 162.0

### Polymorphism –

Refers to the ability of OOPs programming languages to differentiate between entities with the same name efficiently. This is done by Java with the help of the signature and declaration of this entities.Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters or both. Overloading is related to compile-time (or static) polymorphism.



#### Fig 2.4 (a) Fig 2.4 (b)

* 1. **Inheritance:**

It is an important pillar of OOP (Object Oriented Programming). Its the mechanism in java by which one is class is allow to inherit the features (fields and methods) of another class.

Super Class: The class whose features are inherited is known as superclass (or a base class or a parent class).

Sub Class: class that inherits the other class is called as subclass (or a derived class, extended class, or child class). The subclass can add its own fields and the methods in addition to a superclass fields and methods.

Reusability: Inheritance is the concept of “reusability”, when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

The keyword used for an inheritance is extends.

Syntax:

Class derived-class extends base-class

{

//methods and fields

}

### SINGLE INHERITANCE

Single inheritance is when a category inherits properties from a one class only. All the attributes except private members are inherited or extended by child class from parent class.

class A

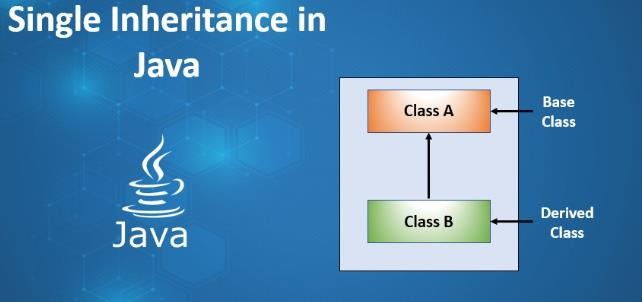
{

}

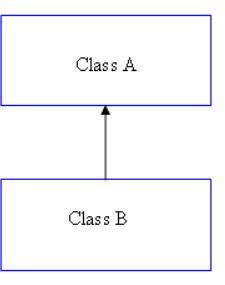
class B extends A

{

}



#### Fig 2.4 (a)

**Fig 2.4 (b)**

* + - **MULTILEVEL INHERITANCE**

Multilevel inheritance is when a category inherits properties from derived class. This derived class becomes the parent of the new child class. It allows accessing of grandparent class attributes by the kid class also.

class A

{

}

class B extends A

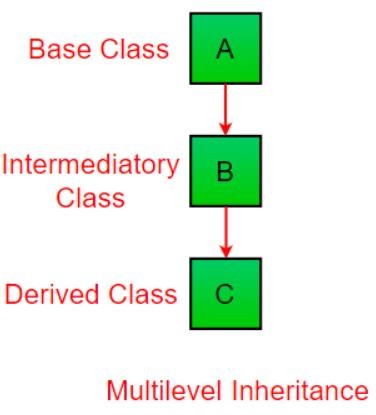
{

}

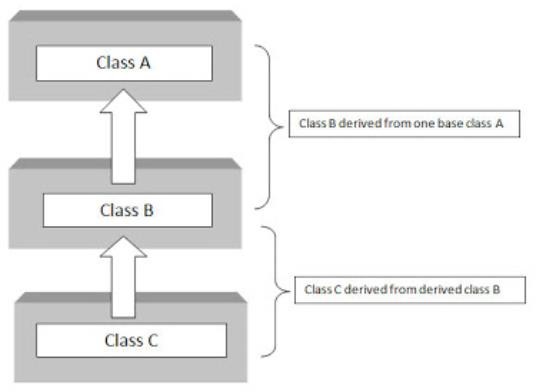
class C extends B

{

}



#### Fig 2.4 (c)



**Fig 2.4 (d)**

* + - **HIERARCHICAL INHERITANCE**

Hierarchical inheritance is when a category is an inherited two or more classes. during this sort of inheritance all of the super class’s sub classes inherit same attributes of the parent class.

class A

{

}

class B extends A

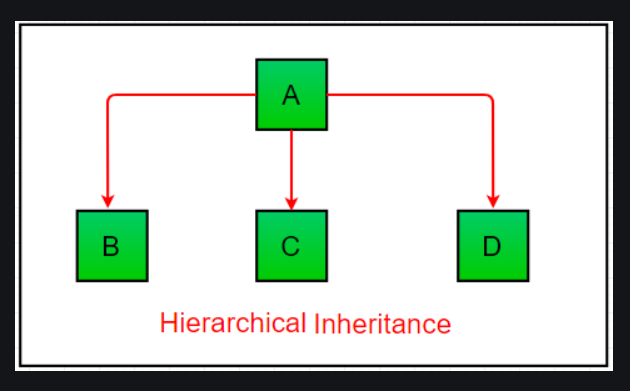
{

}

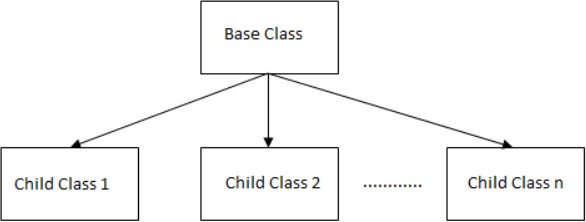
class C extends A

{

}



#### Fig 2.4 (e)



**Fig 2.4 (f)**

## ABSTRACT CLASS

An abstract class may be a template definition of methods and variables of a category which may be a category of objects that contains one or more abstracted methods. Abstract classes are utilized in all OOPs languages. Objects or classes maybe abstracted, which suggests that they're summarized into characteristics that are relevant to the present program’s operation.

Individual instances that are resulting from classes are objects. Declaring a category as abstract means it cannot be directly instantiated, which suggests that an object cannot be created from it. That protects the code from getting used incorrectly. An Abstract class which subclasses are to be further define as attributes are necessary for individual.

Abstract classes similar to the main classes, which are the default type. A concrete class has no abstracted methods and should be instantiated and utilized in code.

### Syntax:

Class abstract classname ,…-

Points to Remember:

* An abstract class should be announced with an abstract keyword.
* It have only abstract methods and also non-abstract methods.
* It can't be instantiated.
* It can have constructors as well as static methods also.
* It can have final methods which may force the subclass to not change the body of the tactic .

## MULTITHREADING

Multithreading in java could also be a process of executing multiple threads. A thread may be a lightweight sub-process which is that the smallest unit of processing. Multiprocessing as well as multithreading, both are used to perform multitasking. However, we use multithreading than multitasking than multiprocessing because threads use a shared memory. They won’t allocate separate memory spacesto saves memory, and switching between the threadswhich takes less time to process. Java Multithreading is typically utilized in games, animation, etc.

Multitasking could also be a process of executing multiple tasks simultaneously. We use multitasking to utilize the CPU. Multitasking are often achieved in two ways:

Process-based Multitasking (Multiprocessing) Thread-based Multitasking (Multithreading)

1. Process-based Multitasking (Multiprocessing)

* Each process has an address in memory. Process based multitasking allocates a separate memory areas.
* A process is heavyweight.
* Cost of communication between the tactic is high.
* Switching from one process to a different requires a while for saving and loading register, memory maps, updating lists, etc.

1. Thread-based Multitasking (Multithreading)

* Threads share an equivalent address space.
* A thread is lightweight.
* Cost of communication between the thread Is slow

## I/O FUNCTIONS IN JAVA

An I/O function in java is wont to process the input and provides output. It uses concept of Streams to operations fast. we will also perform file handling in java using these streams.

* System. Out
* System.in
* System. Err
* Output Stream
* Input Stream Example:

Scanner s=new Scanner (System.in); System.out.println (“Hello”);

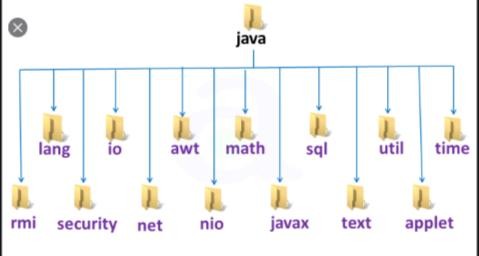
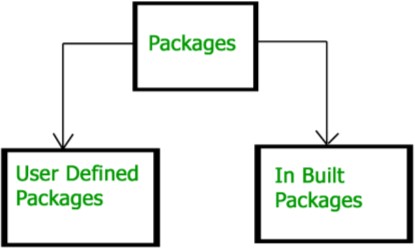
System.err (0);



#### Fig 2.7(a)

* 1. **PACKAGES IN JAVA**

Packages in java is to encapsulate a gaggle of classes sub-packages and interfaces. It prevents naming conflicts and also provides controlled access and are referred to as data encapsulation**.**



#### Fig 2.8 (a) Fig 2.8 (b)

* Packages are containers for classes that are used the category name.
* Through the utilization of the interface keyword ;Java allows to completely abstract the interface from its implementation.
* By Using interface, we will specifies the set of methods which will be implemented by one or more class.
* In interface itself, doesn't actually define any implementation.
* A class can implement quite one interface.,
* Java provides a partitioning the category name space into more manageable chunks. This mechanism is named a package.
* In package is both a naming and a visibility control mechanism.
* It is feasible to define classes inside a package that aren't accessible by code outside that packages.
* we will define class members that are only exposed to other members of an equivalent package.
  + 1. **Defining Package**
* To create a package simply include a package command is that the first statement during a Java source file.
* Any classes declared within that file is belong to the required package.
* package statement defines name space during which classes are stored.
* If we skip the package statement the category names are put into the default package, which has no name.
* The general sort of the package statement is as follows package pkg;
* during this pkg is that the name of the package.
* for instance we will consider the subsequent statement creates a package called My Package. package My Package.
* The general sort of a multileveled package statement is taken into account as: package pkg1[.pkg2[.pkg3]];

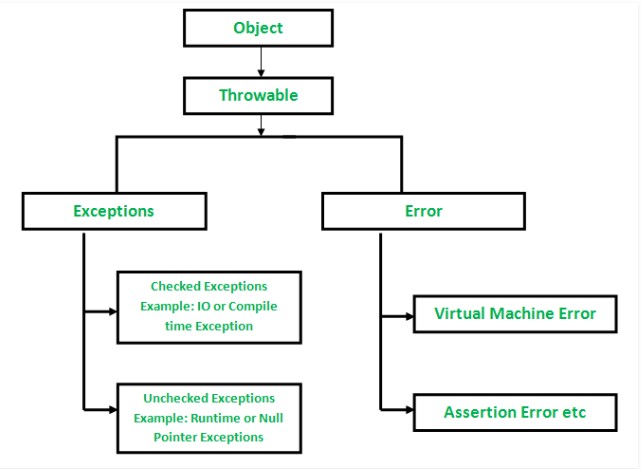
# Exception handling

* It allows us to handle the runtime errors caused by exceptions.
* An exception is an not normal ,It occurs during the execution of a program like compile time that includes the flow of instructions.
* Languages that do not support exception handling, errors must be checked and handled manually—typically through the use of error codes [system generated error codes from 0 to 499].
* This approach is as cumbersome as it is troublesome.
* Java’s exception handling avoids these problems and, brings semantic error management into the object- oriented world. All exception handling types are subclasses of the built-in class which is called Throwable.

### Exception Types:

* + - * Throwable is at the highest of the exception class hierarchy.
      * In the below Throwable they are two subclasses which takes partition exceptions into two different branches.
      * One branch is headed by Exception.
      * This class is employed for exceptional conditions that user programs should catch.
      * This is also the category that you simply will subclass to make your own custom exception types.
      * There is a crucial subclass of Exception, called Runtime Exception. Exceptions of this sort are automatically defined for the programs that you simply write and include things like division by zero and invalid array indexing.
      * The other branch is topped by Error.
      * These are not expected to be caught under normal circumstances by your program, are typically created in response to catastrophic failures that cannot usually be handled by your program.
      * Exceptions of type Error are employed by the Java run-time system to point errors having to try to to with the run-time environment, itself.
      * Stack overflow is an example of such a mistake .

### Exception Hierarchy



#### Fig 2.9(a)

* + 1. **Exception-Handling in Java five keywords**

1. **try**

Program statements that you simply want to watch for exceptions are contained within a try block.

If an exception occurs within the try block, it is thrown.[an object representing that exception is created and thrown in the method that caused it]

### catch

The code which going to can catch this exception (using catch) and handle it in some rational manner.

It generated exceptions which are automatically thrown by the Java run-time application.

### throw

For manually throw an exception, we use the keyword called as throw.

### throws

Any exception that's thrown out of a way must be specified intrinsically by a throws clause.

### finally

Any code that absolutely must be executed after a try block completes, is put during a finally block.

### Encapsulation:

Encapsulation is known as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Another way to think about encapsulation is, a protective shield that prevents the data from being accessed by the code outside this shield.

Technically in encapsulation, the variables of data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared. As in encapsulation, the data in a class is hidden from other classes, so it is also known as data-hiding .Encapsulation can be achieved by Declaring all the variables

in the class as private and writing public methods in the class to set and get the values of variables.

Overriding in Java

In any object-oriented programming language Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of a its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature and same return type as a method in its super-class, then the method in the subclass is said to override the method in the super-class.

Method overriding is one of the way by which java achieve Run Time Polymorphism.The version of a method that is executed will be determined by the object that is used to invoke it. If an object of a parent class is used to check the method, then the version in the parent class will be executed, but if an object of the subclass is used to invoke the method, then the version in the child class will be executed. In other words, it is the type of the object being referred to that determines which version of an overridden method will be executed.

### Data:

Abstraction is the property by virtue of which the essential details are displayed to the user. The trivial or the non-essentials units are not displayed to the user. Ex: A car is seen as a car rather than its individual components. Data Abstraction also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and the behaviors of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car. This is what abstraction is.

In java, abstraction is achieved by interfaces and abstract classes. We can achieve 100% abstraction using interfaces.

### Class:

A class is a user defined blueprint from which objects are created. It represents the set of properties or methods that are common to all objects of one type. class declarations can include these components

Modifiers: A class can be public or has default access

Class name: The name should begin with a initial letter (capitalized by convention).

Superclass : The name of the class’s parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.

Interfaces (): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.

Body: The class body surrounded by braces, { }.

Object: It is a basic unit of Object Oriented Programming and represents as the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of:

State: It is represented by attributes of an object. It also reflects the properties of an object.

Behavior: It is represented by the methods of an object. It also reflects the response of an object with other objects.

Identity: It gives a unique name to an object and the enables one object to interact with other objects

6. A method is a collection of statements that perform some of the specific task and return result to the caller. A method can perform some specific task without returning anything. Methods allow us to reuse the code without retyping the code. In Java, every method must be part of some class which is different from languages like C, C++ and also Python.

Methods are time savers and help us to reuse the code without retyping the code.

Method Declaration

method declarations has six components:

Access Modifier: Defines access type of the method i.e. from where it can be accessed in your application. In Java there 4 type of the access specifiers.

Public: accessible in all class in your application.

Protected: accessible within the package in which it is defined and in its subclass(es)(including subclasses declared of outside the package)

private: accessible only within the class in which it is defined.

default (declared/defined without using modifier): accessible within same class and package within which its class is defined.

The return type: The data type of the value returned by the method or void if does not return a value.

Method Name: the rules for field names apply to method names as well, but the convention is a little different.

Parameter list: Comma separated list of the input parameters are defined, preceded with their data type, within the enclosed parenthesis. If there are no parameters, you must use empty parentheses ().

Exception list: The exceptions you expect by the method can throw, you can specify these exception(s).

Method body: it is enclosed between braces. The code you need to be executed to perform your intended operations.

8. Passing: Objects communicate with one another by sending and receiving information to each other. A message for object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and information to be sent.

**Chapter 3**

**ALGORITHM**

### ALGORITHM

* + - Start.
    - It displays the home page is about to enter the details of username and password.
    - If your pet is a registered user then provide valid details. Then it will go further or else it shows invalid username or password.
    - If your pet is not a registered user then fill up the registration form by providing your pet details.
    - Registration form

1. Name
2. Age
3. Breed
4. Colour
5. Weight
6. Unique features
7. Owner name
   * + For missing pets:
8. Name
9. Age
10. Breed
11. Colour
12. Weight
13. Unique features

**Name**

**Age**

**Colour**

**Weight**

**Enter the pet name**

**Enter the Colour of your pet**

**Enter the weight of your pet**

1. Owner name
2. Pet photo

### Design goals

ONLINE PORTAL FOR PET REGIATRATION

**Registration form**

**Enter the valid username and password**

**Enter the pet’s age**

**Breed**

**Enter the type of your pet**

**Enter the missing pet name**

**Enter the Colour of your missing pet**

**Enter the weight of your missing pet**

**Unique**

**features**

**Enter the unique features of your**

**pet**

**Owner name**

**Enter the owner name**

**Missing pets form**

**Name**

**Age**

**Enter the missing pet’s age**

**Breed**

**Enter the type of your missing pet**

**Colour**

**Weight**

**Unique**

**features**

**Enter the unique features of your**

**missing pet**

**Owner**

**name**

**Enter the owner name**

**Pet photo**

**Upload the missing pet photo**

## CHAPTER 4

**IMPLEMENTATION**

1. The program starts with displaying
   * Username
   * Password

By entering valid details it goes to registration form.

1. Registration form
2. Name
3. Age
4. Breed
5. Colour
6. Weight
7. Unique features
8. Owner’s name

If you are already logged then login to that page.

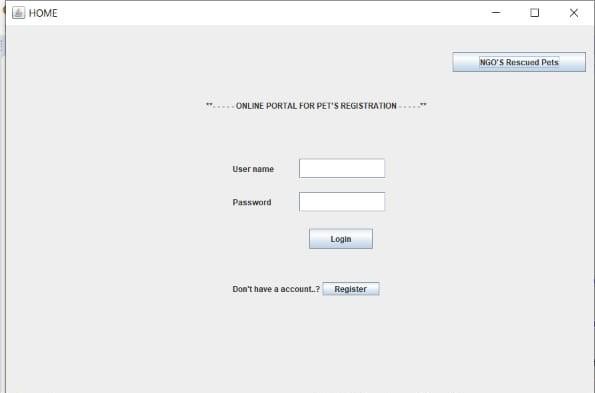
1. For missing pets registration form
   * Name
   * Age
   * Breed
   * Colour
   * Weight
   * Unique features
   * Owner’s name
   * Pet’s photo
2. NGO’s rescued pets

It shows the details of pets which are rescued by NGO’s.

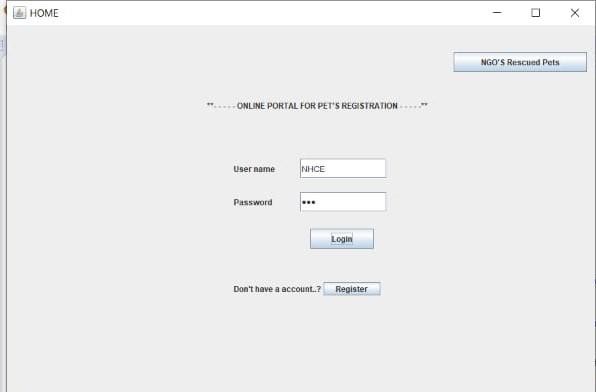
## Chapter 5

**RESULTS**

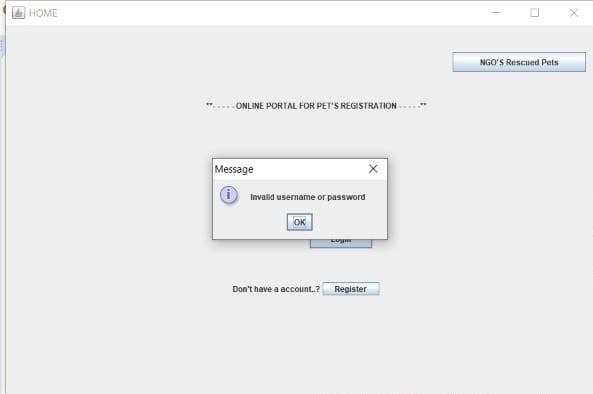
### : Home



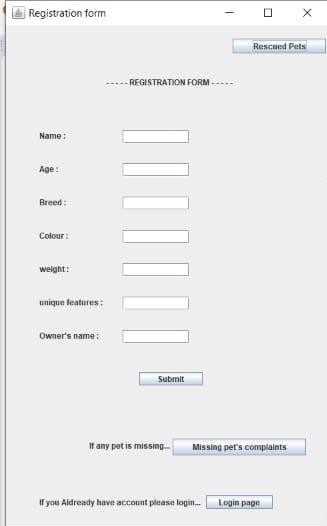
* 1. **: After entering username and password**



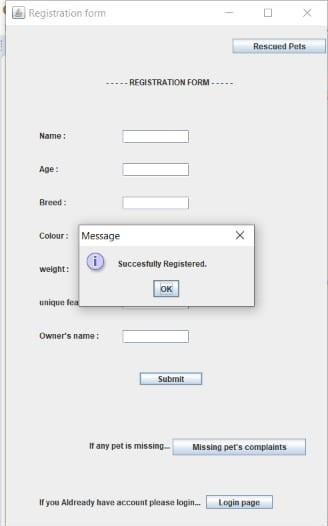
* 1. **:** If we enter invalid username and password



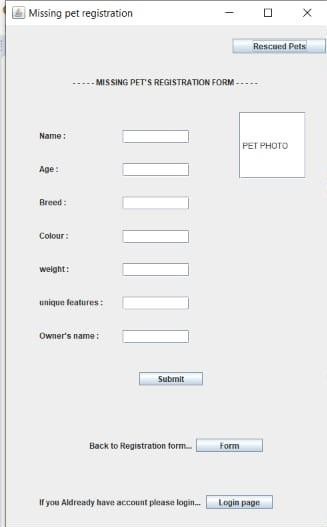
* 1. **:** Registration form



* 1. **:** After entering details in registration form



* 1. **:** Missing pets registration form



* 1. **:** Rescued pets



**CHAPTER 6**

**CONCLUSION**

The project is successfully completed the extent possible. The results of the project are shown earlier.

### Future Scope

The project has the code which is used to create the online portal for the pet shop using the Graphical User Interface. Which is very helpful for each and every customer to login or register the details of his/her dog or they can also file a complaint about the missing dog and they can also see the NGOS rescued dogs. By this portal a customer can fulfill his needs without any difficulties.

Some more functions or modules may be added to project. After studying and understanding Graphic User Interface of java, the inputs and outputs can be improved and implemented using GUI.

# REFERENCES

[1] A. Gersho and R. M. Gray, Vector quantization and signal compression. Massachusetts, USA: Springer Science & Business Media, 1992.

\*2+ R. M. Gray, “Vector quantization,” IEEE ASSP Magazine, pp. 4–29, April 1984.

\*3+ S. Alkhalaf, O. Alfarraj, and A. M. Hemeida, “Fuzzy-VQ image compression based hybrid PSOGSA optimization algorithm,” in IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2015, pp. 1–6.

1. C.-C. Chang, T. S. Nguyen, and C.-C. Lin, “A reversible compression code hiding using SOC and SMVQ indices,” Information Sciences, vol. 300, pp. 85–99, 2015.
2. H.-T. Hu, L.-Y. Hsu, and H.-H. Chou, “Variable-dimensional vector modulation for perceptual-based DWT blind audio watermarking with adjustable payload capacity,” Digital Signal Processing, vol. 31, pp. 115–123, 2014.

\*6+ N. Farvardin, “A study of vector quantization for noisy channels,” IEEE Transactions on Information Theory, vol. 36, no. 4, pp. 799–809, 1990.

\*7+ H. A. S. Leitao, W. T. A. Lopes, and F. Madeiro, “PSO algorithm applied to codebook design for channel-optimized vector quantization,” IEEE Latin America Transactions, vol. 13, no. 4, pp. 961–967, 2015.

\*8+ N. Farvardin and V. Vaishampayan, “On the performance and complexity of channel- optimized vector quantizers,” IEEE Transactions on Information Theory, vol. 37, no. 1, pp. 155–160, 1991.

\*9+ Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” IEEE Transactions on Communications, vol. 28, no. 1, pp. 84–95, 1980.

\*10+ R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, 1995.

\*11+ “Particle swarm optimization,” in Proceedings of the IEEE International Conference on Neural Network, vol. 4, pp. 1942–1948, 1995.

[12] X.-S. Yang, Firefly algorithm, Nature-Inspired Metaheuristic Algorithms. Luniver Press, 2008.