



**NEW HORIZON  
COLLEGE OF ENGINEERING**

Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC  
Accredited by NAAC with 'A' Grade & Accredited by NBA



**A MINI PROJECT  
REPORT**

*for*

*Mini Project in JAVA (19CSE48)*

**MINI CENSUS**

*Submitted by*

**RANGANATH.K**

**1NH19CS739**

**4<sup>th</sup> SEM/E SEC**

*In partial fulfilment for the award of*

*the degree of*

**Bachelor of Engineering**

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**NEW HORIZON  
COLLEGE OF ENGINEERING**

Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC  
Accredited by NAAC with 'A' Grade & Accredited by NBA



## *Certificate*

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

### **MINI CENSUS**

*Submitted in partial fulfilment of the degree of  
Bachelor of Engineering in  
Computer Science and Engineering by*

**RANGANATH.K**  
USN: 1NH19CS739

*DURING*

*EVEN SEMESTER 2020-2021*

*for*

*COURSE CODE: 19CSE48*

  
10/08/2021  
Signature of Reviewer

  
Signature of HOD

### **SEMESTER END EXAMINATION**

*Name of the Examiner*

*Signature with date*

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

# PLAGARISM CERTIFICATE

1NH19CS739\_RANGANATH.K.pdf

## ORIGINALITY REPORT

6%

SIMILARITY INDEX

%

INTERNET SOURCES

6%

PUBLICATIONS

%

STUDENT PAPERS

## PRIMARY SOURCES

- |   |   |      |
|---|---|------|
| 1 | Noel Kalicharan. "Advanced Topics in Java", Springer Science and Business Media LLC, 2014<br>Publication  | 1%   |
| 2 | "A Concise and Practical Introduction to Programming Algorithms in Java", Springer Science and Business Media LLC, 2009<br>Publication  | 1%   |
| 3 | Akshay Bharadwaj K H, Deepak, V Ghanavanth, Harish Bharadwaj R, R Uma, Gowranga Krishnamurthy. "Smart CCTV Surveillance System for Intrusion Detection With Live Streaming", 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2018<br>Publication | 1%   |
| 4 | Joachim Goll, Cornelia Heinisch. "Java als erste Programmiersprache", Springer Science and Business Media LLC, 2014<br>Publication  | 1%   |
| 5 | B. N. Madhukar, Sanjay Jain. "A duality theorem for the discrete sine transform (DST)", 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2015<br>Publication  | 1%   |
| 6 | Jeanne Boyarsky, Scott Selikoff. "Advanced Class Design", Wiley, 2019<br>Publication  | < 1% |
| 7 | Perry Xiao. "Java Programming for Networking Applications", Wiley, 2018<br>Publication  | < 1% |

8	<a href="#">Yang Li</a> , <a href="#">Luhua Zhao</a> , <a href="#">Entong Wang</a> . "Research on Network Drop Truck Scheduling Based on Dijkstra Algorithm", ITM Web of Conferences, 2019 Publication	<1% S
9	<a href="#">Mousumi Das</a> . "Vulnerability to Food Insecurity: A Decomposition Exercise for Rural India using the Expected Utility Approach", Social Indicators Research, 2021 Publication	<1%
10	<a href="#">Vaskaran Sarcar</a> . "Interactive Object Oriented Programming in Java", Springer Science and Business Media LLC, 2016 Publication	<1%
11	"Illustrated C# 2008", Springer Science and Business Media LLC, 2008 Publication	<1%
12	<a href="#">P. Devendran</a> , <a href="#">S. Prasath</a> , <a href="#">Smeya Mohanan</a> , <a href="#">G. Praveen</a> . "Articulated robotic system-wireless, manual and shadow mode", Materials Today: Proceedings, 2021 Publication	<1%

Exclude quotes Off  
Exclude bibliography Off

Exclude matches Off

## **ABSTRACT**

Here I am going to implement the MINI CENSUS which will be the very helpful topic to collect village census and in this project it will collect the information about the village based on each houses, in that house how many people are there and their ages, contact details etc. like that it will collect and store, in future by modifying some code it can also be used in various fields like surveys and now days we are get vaccinating so based on these we can also collect the people are vaccinated or not.

Mini census mini project is used to collect village's census and here I am going to implement it in java using encapsulation, inheritance, abstraction etc. So using these we can collect, modify, store, delete and display of census we collected and it's secured with password protection.

**SCOPE:** Our project includes

- keeps the village data
- Add new house details
- Modify the house details
- Display and search the selected house details

# 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, New Horizon College of Engineering, for his constant support and encouragement.

I am grateful to **Dr. Amarjeet Singh**, 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 also express my gratitude to **Ms Dr. Vaishali M Deshmukh**, Associate professor, Department of Computer Science and Engineering, 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.

**RANGANATH.K**  
**USN: 1NH19CS739**

## **INDEX**

<b>Sl.NO.</b>	<b>TITLE</b>	<b>PAGE NO</b>
	<b>ABSTRACT</b>	<b>I</b>
	<b>ACKNOWLEDGEMENT</b>	<b>II</b>
	<b>LIST OF FIGURES</b>	<b>V</b>
<b>1.</b>	<b>INTRODUCTION</b>	
1.1.	PROBLEM DEFINITION	1
1.2.	OBJECTIVES	1
1.3.	METHODOLOGY TO BE FOLLOWED	2
1.4.	EXPECTED OUTCOMES	2
<b>2.</b>	<b>FUNDAMENTALS OF JAVA PROGRAMMING</b>	
2.1.	INTRODUCTION	3 - 4
2.2.	CLASS	4
2.3.	OBJECT	5
2.4.	INHERITANCE	5 - 6
2.5.	POLYMORPHISM	6
2.6.	ABSTRACT CLASS	7
2.7.	ENCAPSULATION	8
2.8.	EXCEPTION HANDLING	9

<b>3. REQUIREMENTS AND SPECIFICATION</b>	
3.1. HARDWARE REQUIREMENTS	<b>10</b>
3.2. SOFTWARE REQUIREMENTS	<b>10</b>
<b>4. DESIGN</b>	
4.1. DESIGN GOALS	<b>11</b>
4.2. FLOWCHART/AGLORITHM	<b>12 - 13</b>
<b>5. IMPLEMENTATION</b>	
5.1. MODULE 1 FUNCTIONALITY	<b>14 - 15</b>
5.2. MODULE 2 FUNCTIONALITY	<b>16 - 17</b>
5.3. MODULE 3 FUNCTIONALITY	<b>18 - 19</b>
5.4. MODULE 4 FUNCTIONALITY	<b>20 - 21</b>
5.5. MODULE 5 FUNCTIONALITY	<b>22 - 24</b>
<b>6. RESULTS</b>	
6.1. MAIN SCREEN	<b>25</b>
6.2. LOGIN AND LOGOUT	<b>25</b>
6.3. CREATION OF DETAILS	<b>26</b>
6.4. MODIFICATION AND DELETION	<b>27 - 28</b>
6.5. SEARCHING AND GOVERNMENT SCHEMES	<b>28 - 30</b>
6.6. DISPLAYING	<b>30 -31</b>
<b>7. CONCLUSION</b>	<b>32</b>
<b>REFERENCES</b>	<b>33</b>



## **LIST OF FIGURES**

<b><u>Figure No</u></b>	<b><u>Figure Description</u></b>	<b><u>Page No</u></b>
2.1.1	Features of java	03
2.1.2	Concepts in java	04
2.4.1	Types of inheritance	06
2.5.1	Types of polymorphism	06
2.8.1	Exception Hierarchy	09
4.1.1	Designing goals of project	11
6.1.1	Menu screen	25
6.2.1	Login and Logout	25
6.3.1	Creation of details	26
6.4.1	Modification of details	27
6.4.2	Deletion of details	28
6.5.1	Searching of details	29
6.5.2	Government schemes	29 - 30
6.6.1	Displaying	30 - 31

# CHAPTER 1

## INTRODUCTION

### 1.1 PROBLEM DEFINITION

Develop a mini census project to collect the census information in the village level and needs to display the menu for the user and the menu contains create, delete, modify, search, display and exit options and in this program admin need to login to access the create, modify, delete options so data have secured and a display, search options are made it available publicly so anyone can access this two options.

This project will be the very helpful topic to collect a village census and in this project it will collect the information about the village based on each house, in that house how many people are there and their ages, contact details etc. like that it will collect and store.

Here this project will be developed using a java using the concepts of a java like an encapsulation, polymorphism, inheritance and etc.so using these concepts we can easily store and access the data of a mini census.

### 1.2 OBJECTIVES

- The main objective of developing this project is get know about java and their concepts like class, objects, inheritance, polymorphism, encapsulation, abstraction, etc.
- By developing mini projects students can be capable of solving a real time problems related to computer science field.
- This projects helps to get more knowledge about java by implementing it in own way.
- The objective of developing such a computerized program reduces the paper work and it's safe.

### 1.3 METHODOLOGY TO BE FOLLOWED

- In this project we need follow some methodology while dealing with this project.
- In the main screen you can select only three options for operations and exit option is made for completely closing the output screen.
- After selecting the login option you must remember the last saved admin password, so after entering the admin password only it will allow doing the further process.
- This project is developed using the eclipse so its show better performance in eclipse.

### 1.4 EXPECTED OUTCOMES

- In the main screen it is going to display welcome to MINI CENSUS and the menu for user selection and the menu contain the following list.
  1. Login
  2. Display –it displays all data stored in the mini census.
  3. Search –display only searched house details.
  4. Government schemes-list the schemes of government
  5. Exit -exit from the screen.
- In the login option, it will ask user to enter the admin password
- After entering admin password ,if it matches then it will show a further menu to admin.
  1. Create new –entering the new details
  2. Modify details –modify the existing details
  3. Delete details –deleting the details from the mini census
  4. Logout –it will get back into the main screen.

## CHAPTER 2

# FUNDAMENTALS OF JAVA PROGRAMMING

## 2.1 INTRODUCTION

Java programming language was developed by Sun Microsystems of the USA in 1991, it had been originally called Oak by James Gosling, who was one in every of the inventors of the language. The most goals for the developers were to form the language highly reliable, portable and straightforward.

The team for the event of Java language included Patrick Naughton, who discovered that the existing languages like C and C++ had some major drawbacks in terms of reliability and portability. They modelled the new language Java on C and C++ while removing some features which they considered as constraints. This made Java a very simple, portable and powerful language. Java is an object-oriented programming language with its runtime environment. It's a mixture of features of C and C++ with some essential additional concepts. Java is compatible for both standalone and web application development and is meant to produce solutions to most of the issues faced by users of the web era.

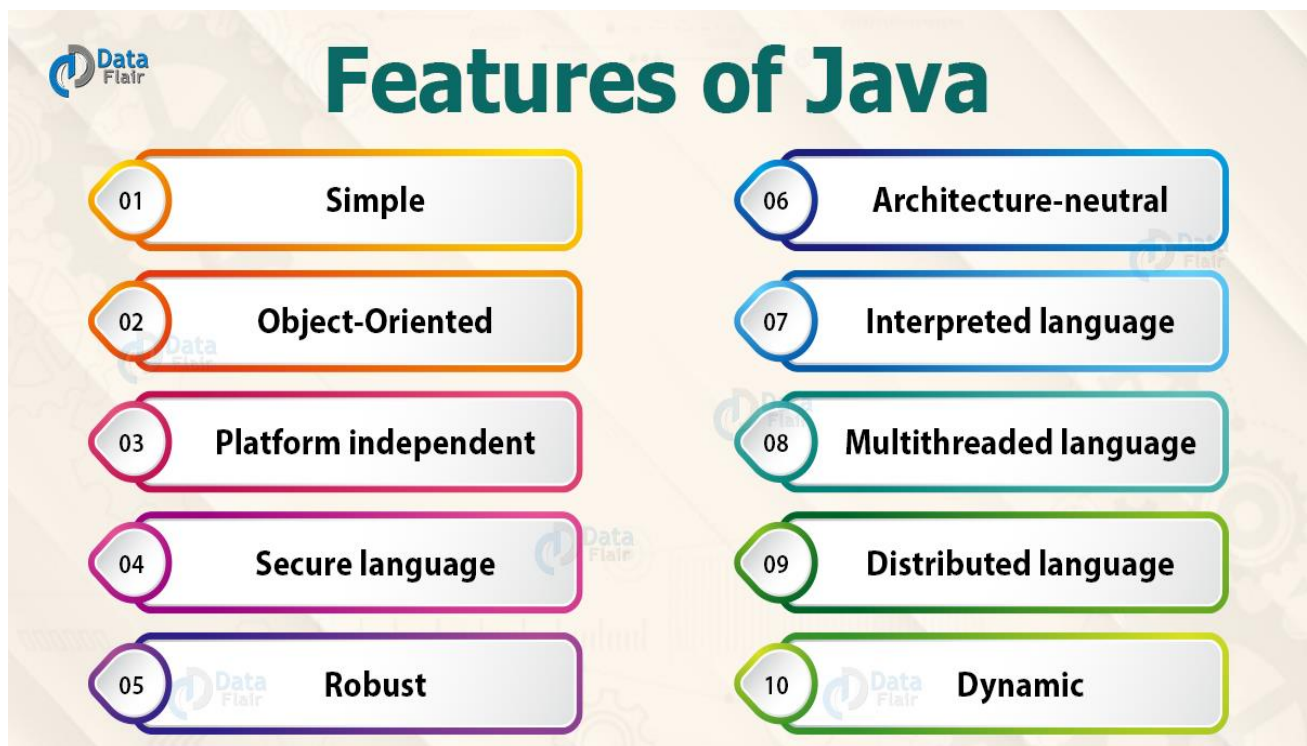
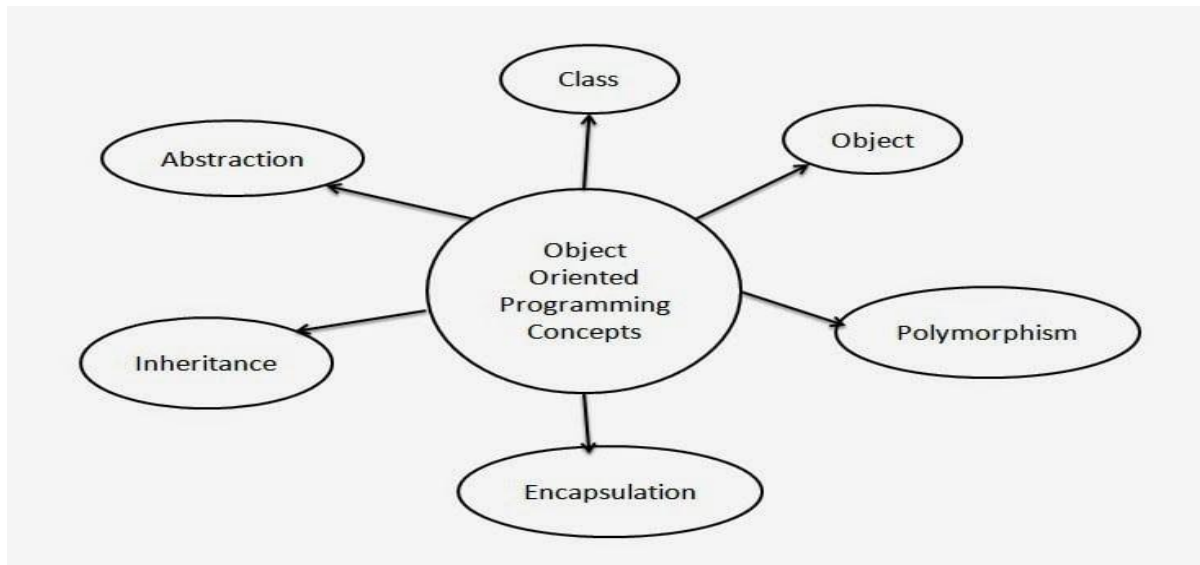


Fig 2.1.1 Features of java.



**Fig 2.1.2 Concepts in java.**

## 2.2 CLASS

Class are a blueprint or a group of instructions to create a selected sort of object. It is a basic concept of Object-Oriented Programming which revolves round the real-life entities. Class in Java determines how an object will behave and what the thing will contain.

### Syntax

```
class <class_name>
{
    field;
    method;
}
```

### Example:

```
class Student.{
    String name;
    int rollno;
    int age;
    void info(){ // some code
    }
}
```

### 2.3 OBJECT

Object is an instance of a class. An object in OOPS is nothing but a self-contained component which consists of methods and properties to form a specific sort of data useful. For example colour name, table, bag, barking. When you send a message to an object, you're asking the thing to invoke or execute one among its methods as defined within the class.

From a programming point of view, an object in OOPS can include a data structure, a variable, or a function. It has a memory location allocated. Java Objects are designed as class hierarchies.

#### Syntax

```
ClassName ReferenceVariable = new ClassName();
```

**Example:** Object creation

```
Student std = new Student();
```

### 2.4 INHERITANCE

Inheritance may be a mechanism during which one class acquires the property of another class. In Java, when an "Is-A" relationship exists between two classes, we use Inheritance. The parent class is called a super class and the inherited class is called a subclass. The keyword `extends` is employed by the sub class to inherit the features of super class. Inheritance is vital since it results in the reusability of code.

#### Java Inheritance Syntax:

```
class subClass extends superClass
{
    //methods and fields
}
```

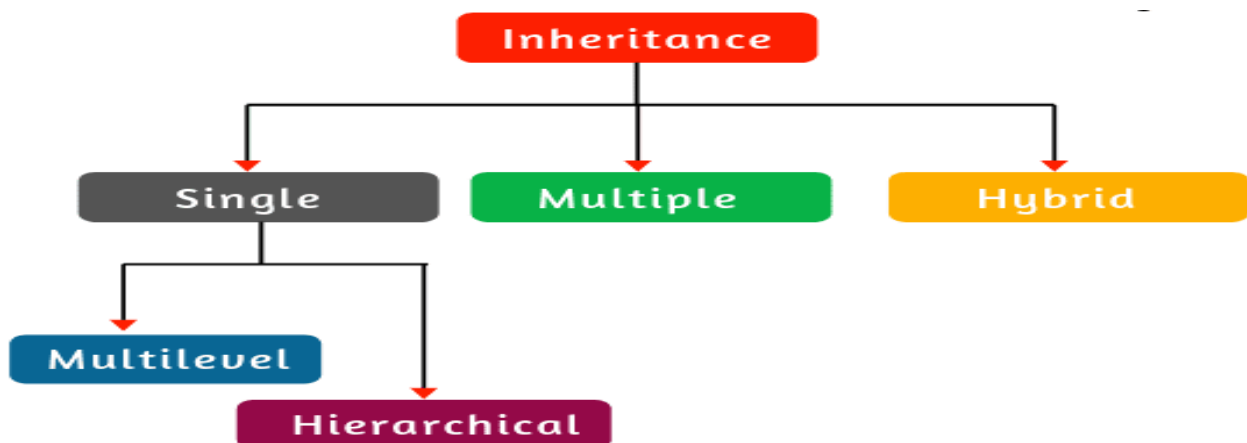


Fig 2.4.1 types of inheritance.

## 2.5 POLYMORPHISM

Polymorphism is one of the OOPs features that allow us to perform a single action in different ways. Polymorphism is that the capability of a way to try to do various things supported the object that it's acting upon. In other words, polymorphism allows you define one interface and have multiple implementations.

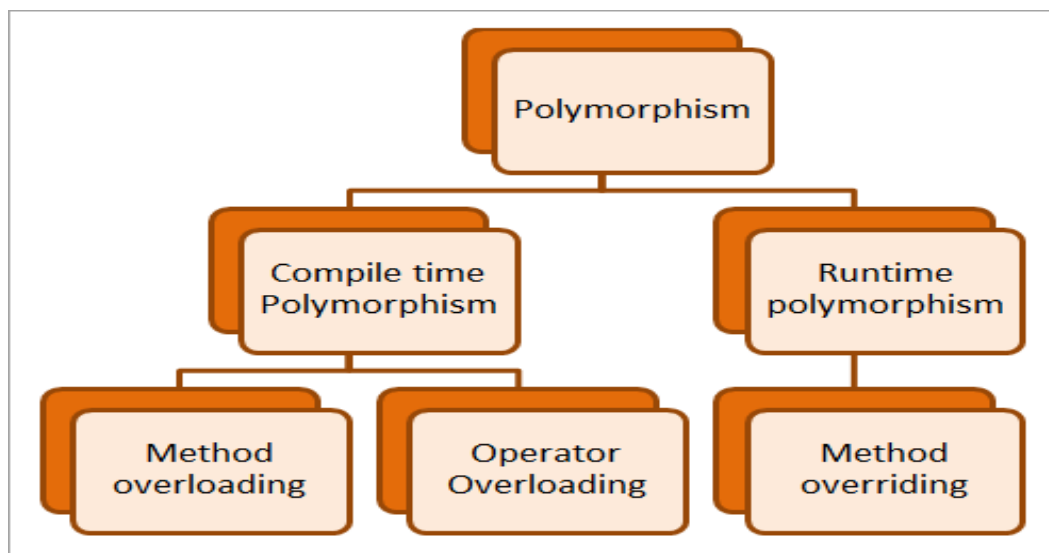


Fig 2.5.1 Types of polymorphism.

### 2.6 ABSTRACT CLASS

Abstraction is the concept of object-oriented programming that "shows" only essential attributes and "hides" unnecessary information. The main purpose of abstraction is hiding the unnecessary details from the users.

Abstraction is selecting data from a larger pool to show only relevant details of the object to the user. It helps in reducing programming complexity and efforts. It is one of the most important concepts of OOPs.

#### **Syntax:**

```
<Access_Modifier> abstract class <Class_Name> {  
  
    //Data_Members;  
  
    //Statements;  
  
    //Methods  
  
}
```

#### **Abstract Class:**

Abstract Class is one of class in OOPs that declare one or more abstract methods. These classes can have abstract methods also as concrete methods. A normal class cannot have abstract methods. An abstract class have a class that contains a minimum of one abstract method.

#### **Abstract Method:**

Abstract Method is a type of method that has just the method definition but doesn't contain implementation. A method without a body is called as an Abstract Method. It must be declared in an abstract class. The abstract method will never be final because the abstract class must implement all the abstract methods.

#### **Advantages of Abstraction**

- The main advantage of using an Abstraction in Programming is that it allows you to group several related classes as siblings.
- Abstraction in Object Oriented Programming helps to scale back the complexity of the planning and implementation process of software.



### 2.7 ENCAPSULATION

Encapsulation in Java is one of the powerful mechanisms for storing the data members and data methods of a any class together. It is wiped out the shape of a secure field accessible by only the members of that class

Encapsulation in Java is an object-oriented procedure of mixing the data members and data methods of the class inside the user-defined class. It is important to declare this class as private

#### Syntax:

```
<Access_Modifier> class <Class_Name>

{

private <Data_Members>;

private <Data_Methods>;

}
```

#### Advantages of Encapsulation

- Encapsulation is binding the info with its connected functionalities. Here functionalities mean "methods" and information suggests that "variables"
- Therefore we have a tendency to keep variable and ways in one place. That place is "class." class is that the bottom for encapsulation.
- With Java Encapsulation, you'll be able to hide (restrict access) to important data members in your code, that improves security
- If data member is said "private", then it will solely be accessed at intervals a similar class. No outside class will access data member (variable) of alternative class.
- However, if you would like to access these variables, you've got to use public "getter" and "setter" ways.

### 2.8 EXCEPTION HANDLING

Exception in Java is an occasion that interrupts the execution of program instructions and disturbs the traditional flow of program execution. It is an object that wraps mistake event information that occurred within a way and it's passed to the runtime system. In Java, exceptions are mainly used for indicating differing types of error conditions.

#### Syntax:

```
class Exception{  
public static void main(String args[]){  
try{  
//code that may raise exception  
}  
catch(Exception e){  
// rest of the program  
}}}
```

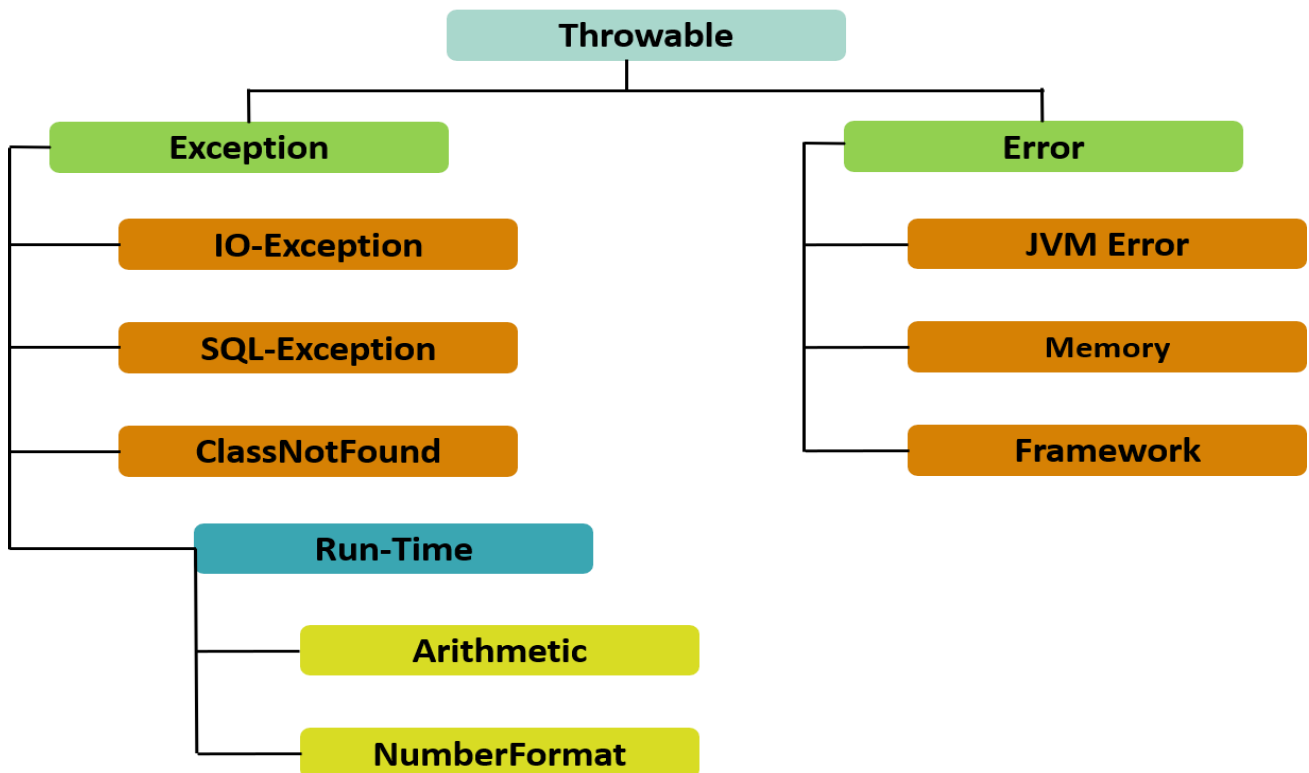


Fig 2.8.1 Exception Hierarchy.

## **CHAPTER 3**

# **REQUIREMENTS AND SPECIFICATION**

### **3.1 HARDWARE REQUIREMENTS**

- A laptop/Desktop
- Windows 7/8/10,linux or any operating system
- 500 MB or higher RAM
- Minimum 1 GB ROM.
- Keyboard, Mouse and Monitor.

### **3.2 SOFTWARE REQUIREMENTS**

- Pre-installed JDK with all its components like JRE,JVM.
- Eclipse software
- Notepad and Command prompt.

## CHAPTER 4

# DESIGN

### 4.1 DESIGN GOALS

- Allow users to easily see and make it user friendly.
- Give users an easy way to back out (i.e. change & error recovery).
- Allow user to complete their task without being distracted by software or losing train of thought, which is while they are reading and typing.
- Everything will be made it by menu selection for easy understand.
- Give users access to information they need to complete their task.
- Wishing the user whenever they open the screen.
- Collects the details one by one without overloading.



**Fig 4.1.1 Designing goals of project.**

### 4.2 ALGORITHM

1. Start.
2. Display WELCOME TO MINI CENSUS and menu for user selection.
3. Options are as follows.

- A. Login
- B. Display
- C. Search
- D. Government schemes
- E. Exit

A. Login( ) option is made using **encapsulation** ,after selecting the login option then it asks user to enter the admin password.

if password matches it will allows to go further selections and they are.

- i. Create new
- ii. Modify details
- iii. Delete details
- iv. Logout.

i. If you select the Create new( ) option it going collects the village name, House number, number of peoples and peoples name ,age ,gender.

ii. If you select the Modify details( ) option it asks user to enter house number for search and modify.

If(house number matches)

- it will again going to collects information of respected house and it will be implemented using **inheritance**

else

Print searched house not found to modify.

iii. If you select the Delete details( ) option it asks user to enter the house number for deleting entire house details.

iv. Logout( ) is for getting back to the main menu.

- B. Display( ) is the option available for everyone, so it list the all details stored in the mini census by columns with different sections.
- C. Search( ) option it asks user to enter the house number for search and display the respected house details.
- D. Government schemes( ) this option ask user to enter the age for displaying the schemes available for that age.
- E. Exit( ) option is for exit from the screen.

4. End

## CHAPTER 5

# IMPLEMENTATION

### 5.1 MODULE 1 FUNCTIONALITY

➤ Class for linked list and creation of details.

In this module I am going to declare class for singly linked list and there is one inner class for creating nodes in linked list, and other class create is for getting the data from user and stores in the nodes of linked list.

```
class LinkedList
{
    census head;

    static class census
    {
        String village;
        int houseno;
        int num;
        String name[]=new String[10];
        int age[]=new int[10];
        String gender[]=new String[10];
        census next;

        census(String vil,int hn,String nm[],int ag[],String gen[],int n )
        {
            num=n;
            village=vil;
            houseno=hn;

            for(int i=0;i<n;i++)
            {
                name[i]=nm[i];
                age[i]=ag[i];
                gender[i]=gen[i];
            }
        }
    }
}
```

## Mini Census

---

```
class Create extends LinkedList
{
    Scanner s=new Scanner(System.in);

    LinkedList insert(LinkedList list)
    {
        System.out.println("Enter your village name:");
        String vil=s.nextLine();
        System.out.println("Enter the house number");
        int hn=s.nextInt();
        System.out.printf("Enter the number of peoples in that house\n");
        int n=s.nextInt();
        String nm[]=new String[n];
        int ag[]=new int[n];
        String gen[]=new String[n];

        for(int i=0;i<n;i++)
        {
            System.out.println("enter the "+(i+1)+" person name");
            nm[i]=s.next();
            System.out.printf("Enter the age\n");
            ag[i]=s.nextInt();
            System.out.printf("Enter the gender\n");
            gen[i]=s.next();
        }

        census node=new census(vil,hn,nm,ag,gen,n );
        node.next=null;

        if(list.head==null)
        {
            list.head=node;
        }
        else
        {
            census last=list.head;
            while(last.next!=null)
            {
                last=last.next;
            }
            last.next=node;
        }
        System.out.println("Deatails saved");
        return list;
    }
}
```



### 5.2 MODULE 2 FUNCTIONALITY

#### ➤ Class for modify and deleting of details

In this module two classes are present, one is modify which will be used for modification of any house details if it's wrong, by entering the house number and another class delete is for deleting the house details by entering the house number.

```
class Modify extends LinkedList
{
    Scanner s=new Scanner(System.in);
    census cur;
    LinkedList modi(LinkedList list)
    {
        int key;
        census cur=list.head;
        System.out.println("Enter the house number to modify");
        key=s.nextInt();
        while(cur!=null&&cur.houseno!=key)
        {
            cur=cur.next;
        }

        if(cur==null)
        {
            JOptionPane.showMessageDialog(null,"Searched house number not present");
            System.out.println("Searched house number not present");
        }

        else if(cur.houseno==key)
        {
            System.out.println("Enter your village name:");
            cur.village=s.next();
            System.out.printf("Enter the number of peoples in that house\n");
            cur.num=s.nextInt();
            for(int i=0;i<cur.num;i++)
            {
                System.out.println("enter the "+(i+1)+" person name");
                cur.name[i]=s.next();
                System.out.printf("Enter the age\n");
                cur.age[i]=s.nextInt();
                System.out.printf("Enter the gender\n");
                cur.gender[i]=s.next();
            }
        }

        return list;
    }
}
```

```
class Delete extends LinkedList
{
    Scanner s=new Scanner(System.in);

    census cur,pre;

    LinkedList del(LinkedList list)
    {
        int key;

        try {
            census cur=list.head;

            System.out.println("Enter the house number to Delete");
            key=s.nextInt();

            if(cur.houseno==key)
            {
                list.head=cur.next;
            }
            else
            {
                while(cur!=null&&cur.houseno!=key)
                {
                    pre=cur;
                    cur=cur.next;
                }
                if(cur==null)
                {
                    System.out.println("Searched house number not present");
                }
                else if(cur.houseno==key)
                {
                    pre.next=cur.next;
                }
            }

            }catch(NullPointerException e)
            {
                System.out.println("Mini census is empty");
            }

            return list;
        }
    }
```

### 5.3 MODULE 3 FUNCTIONALITY

#### ➤ Class for login and searching of details

In this module two classes are present, one is login so this class is programmed using encapsulation and here user must enter the password to access the other information available within that class and another class search is for searching the house details by entering the house number.

```
class Login
{
Scanner s=new Scanner(System.in);
private int pas=1234;
private int pas1;
int choice;

LinkedList password(LinkedList list)
{
System.out.println("Enter the admin password to login");
pas1=s.nextInt();
if(pas1==pas)
{
System.out.printf("_____\\n");
System.out.println("***WELCOME TO ADMIN PORTAL***");
System.out.printf("_____\\n");
do
{
System.out.printf("\\n1:Create\\n2:Modify\\n3>Delete\\n4:Logout\\n");
System.out.println("select your choice:");
choice=s.nextInt();
switch(choice)
{
case 1:
Create cr=new Create();
list=cr.insert(list);
break;
case 2:
Modify m=new Modify();
list=m.modi(list);
break;
case 3:
Delete dl=new Delete();
list=dl.del(list);
break;
case 4:
break;
default:
System.out.println("Enter the valid choice");
}
}
```

## Mini Census

---

```
        }while(choice!=4);
    }
else
{
    System.out.println("Password not matching");
}
return list;
}
}
```

```
class search extends LinkedList
{
    Scanner s=new Scanner(System.in);
    census cur;
void sear(LinkedList list)
{
    int key;
    census cur=list.head;
    System.out.println("Enter the house number to search");
    key=s.nextInt();
    while(cur!=null&&cur.houseno!=key)
    {
        cur=cur.next;
    }
    if(cur==null)
    {
        System.out.println("Searched house number not present");
    }
    else if(cur.houseno==key)
    {
        System.out.printf("_____\n");
        System.out.printf("village\t House no Names \tAge Gender\n");

        System.out.printf("_____\n");

        for(int i=0;i<cur.num;i++)
        {
            System.out.printf(cur.village+"\t"+cur.houseno+"\t "+cur.name[i]+"
\t"+cur.age[i]+" "+cur.gender[i)+"\n");
        }
    }
}
}
```

### 5.4 MODULE 4 FUNCTIONALITY

➤ Class for displaying of all the details.

In this module I am going to declare class for displaying of all the details stored in the mini census and it's going to calculate the all the counts like men's count, women's count and number of houses etc. while displaying.

```
class Display extends LinkedList
{
    private int total,lowage,midage,highage;

    private int numhouse,ml,fm;

    boolean a,b;

void disp(LinkedList list)
{
    census cur=list.head;

    System.out.printf("_____\\n");

    System.out.printf("village\t House no Names \tAge Gender\\n");

    System.out.printf("_____\\n");
    while(cur!=null)
    {
        for(int i=0;i<cur.num;i++)
        {
            System.out.printf(cur.village+"\t"+cur.houseno+"\t "+cur.name[i]+"
\t"+cur.age[i]+" "+cur.gender[i]+"\\n");

            if(cur.age[i]<18)
            {
                lowage+=1;
            }

            else if(cur.age[i]>=18&&cur.age[i]<=65)
            {
                midage+=1;
            }

            else
            {
                highage+=1;
            }

            if(a=(cur.gender[i].startsWith("m")))
            {
```

```
        if(a==true)
        {
            ml++;
        }
    }

    if(b=(cur.gender[i].startsWith("f")))
    {
        if(b==true)
        {
            fm++;
        }
    }

}

total=total+cur.num;

numhouse+=1;

cur=cur.next;

System.out.println("    _*_    _*_    _*_    ");
}

System.out.println("*Total number of houses:"+numhouse);
System.out.println("*Total number of peoples:"+total);
System.out.println("*Number of males:"+ml);
System.out.println("*Number of females:"+fm);
System.out.println("*Number of peoples below 18 years:"+lowage);
System.out.println("*Number of peoples between (18-65) years:"+midage);
System.out.println("*Number of peoples above 65 years:"+highage);
}
}
```

### 5.5 MODULE 5 FUNCTIONALITY

➤ Class for government schemes and driver class

In this module I am going to declare class for government schemes which will be display a schemes available based on user entered age and its programmed using abstraction and another driver class for controlling the whole program and displaying menu, so user can select any option by entering the number.

class myscheme implements Scheme

```
{

    void myage()
    {
        int age;

        System.out.printf("Enter your age\n");
        age=s.nextInt();

        if(age<=18)
        {
            below18();
        }

        else if(age>18)
        {
            above18();
        }

        System.out.printf("***THANK YOU FOR VISTING***\n");
    }

    public void below18()
    {
        System.out.println("\nGovernement Schemes available under this section are:");

        System.out.println("1.Beti Bachao Beti Padhao\n"
            + "2.Sukanya Samriddhi Yojana\n"
            + "3.Balika Samridhi Yojana\n"
            + "4.Mukhyamantri Rajshri Yojana\n"
            + "5.Mukhyamantri Laadli Yojana\n"
            + "6.CBSE Udaan Scheme \n"
            + "7.National Scheme of Incentives to Girls for Secondary
Education\n"
            + "8.Mukhyamantri Kanya Suraksha Yojana\n"
            + "9.Mazi Kanya Bhagyashree Scheme\n")
    }
}
```

```
        + "10.Nanda Devi Kanya Yojana");
    }

    public void above18()
    {
        System.out.println("\nGovernment Schemes available under this section are:");

        System.out.println("1.Pradhan Mantri Jan Dhan Yojna (PMJDY)\n"
            + "3.Rajiv Gandhi Jeevandayee Arogya Yojana (RGJAY)\n"
            + "4.Pradhan Mantri Jeevan Jyoti Bima Yojna (PMJJBY)\n"
            + "5.Rashtriya Swasthya Bima Yojana (RSBY)\n"
            + "6.National Social Assistance scheme\n"
            + "7.Pradhan Mantri Mudra Yojana (PMMY)\n"
            + "8.PM-KISAN (Pradhan Mantri Kisan Samman Nidhi) Scheme\n"
            + "9.Pradhan Mantri Kisan Pension Yojana\n"
            + "10.Pradhan Mantri Shram Yogi Maan-dhan (PM-SYM)\n"
            + "11.Varishtha Pension Bima Yojana (VPBY)\r\n"
            + "12.PRIME MINISTER'S SPECIAL SCHOLARSHIP SCHEME –
PMSSS");
    }
}

public class first
{
    public static void main(String[] args)
    {
        int ch=0;

        LinkedList list=new LinkedList();

        Scanner s=new Scanner(System.in);

        System.out.printf("_____\\n");
        System.out.println("***WELCOME TO MINI CENSUS***");
        System.out.printf("_____\\n");

        do
        {
            System.out.printf("\n1:Login\n2:Display\n3:Search\n4:Government
schemes\n5:Exit\n");

            System.out.println("select your choice:");
            ch=s.nextInt();
        }
    }
}
```



```
        switch(ch)
        {
        case 1:
            Login lg=new Login();
            list=lg.password(list);
            break;

        case 2:
            Display ds=new Display();
            ds.disp(list);
            break;

        case 3:
            search sc=new search();
            sc.sear(list);
            break;

        case 4:
            myscheme sh=new myscheme();
            sh.myage();
            break;

        case 5:
            System.out.println("*****THANK YOU*****");
            break;

        default:
            System.out.println("Enter the valid choice");

        }
    }while(ch!=5);
}
```

## CHAPTER 6

# RESULTS

### 6.1 Main screen

In the below image it displays main menu for user selection.

```
***WELCOME TO MINI CENSUS***

1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
```

Fig 6.1.1 Menu screen.

### 6.2 Login and logout

In the below image if user wants to access the create, modify and delete options so they must enter the admin password to login, after all work is done they can easily logout by selecting the option 4.

```
1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
1
Enter the admin password to login
1234

***WELCOME TO ADMIN PORTAL***

1:Create
2:Modify
3>Delete
4:Logout
select your choice:
4
Your successfully logout
```

Fig 6.2.1 Login and logout.

### 6.3 Creation of details.

In the below image user selected the create option and program going to collecting all information from user and print details saved.

```
***WELCOME TO ADMIN PORTAL***

1:Create
2:Modify
3>Delete
4:Logout
select your choice:
1
Enter your village name:
Rampura.
Enter the house number
101
Enter the number of peoples in that house
4
enter the 1 person name
Ravindra.CH
Enter the age
43
Enter the gender
male
enter the 2 person name
Geetha.KP
Enter the age
37
Enter the gender
female

enter the 3 person name
Akash.RH
Enter the age
14
Enter the gender
male
enter the 4 person name
Amrutha.RH
Enter the age
10
Enter the gender
female
Deatails saved
```

Fig 6.3.1 creation of details.

### 6.4 Modification and deletion

In the below image user selected option two for modifying details, so first it asks user to enter the house number for which house they want to modify the details, then after entering the all details correctly it prints details modified successfully.

```
1:Create
2:Modify
3>Delete
4:Logout
select your choice:
2
Enter the house number to modify
101
Enter your village name:
Hirekabbar.
Enter the number of peoples in that house
4
enter the 1 person name
Ravindra.CH
Enter the age
46
Enter the gender
male
enter the 2 person name
Geetha.KP
Enter the age
39
Enter the gender
female

enter the 3 person name
Akash.RH
Enter the age
14
Enter the gender
male
enter the 4 person name
Amrutha.RH
Enter the age
10
Enter the gender
female
```

Fig 6.4.1 Modification of details.

In the below image user selected option three for deleting details, so first it asks user to enter the house number for which house they want to delete the details, then if the house number present delete and print details deleted successfully else print house number not found.

```
1:Create
2:Modify
3>Delete
4:Logout
select your choice:
3
Enter the house number to Delete
101
House details deleted successfully

1:Create|
2:Modify
3>Delete
4:Logout
select your choice:
3
Enter the house number to Delete
105
Searched house number not present

1:Create
```

Fig 6.4.2 Deletion of details.

## 6.5 Searching and government schemes

In the fig 6.5.1 user selected option three for searching details, so first it asks user to enter the house number for which house they want to search the details, if the searched house number found then display the details else print searched house number not found.

the fig 6.5.2 shows the government schemes, so this option is made for users to search the what are all government schemes available for their age, if user the selects the option 4 it will redirects to the government scheme option, their it first asks user to enter the age so based on age it shows government schemes.

```
1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
3
Enter the house number to search
101
```

village	House no	Names	Age	Gender
Honnali.	101	Rakesh.M	46	male
Honnali.	101	Savitha.R	41	female
Honnali.	101	Srujan.RM	21	male
Honnali.	101	Pooja.RM	18	female

```
1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
3
Enter the house number to search
105
Searched house number not present
```

Fig 6.5.1 searching of details.

```
1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
4
Enter your age
14

Governement Schemes availabile under this section are:
1.Beti Bachao Beti Padhao
2.Sukanya Samriddhi Yojana
3.Balika Samridhi Yojana
4.Mukhyamantri Rajshri Yojana
5.Mukhyamantri Laadli Yojana
6.CBSE Udaan Scheme
7.National Scheme of Incentives to Girls for Secondary Education
8.Mukhyamantri Kanya Suraksha Yojana
9.Mazi Kanya Bhagyashree Scheme
10.Nanda Devi Kanya Yojana
***THANK YOU FOR VISTING***
```

```
1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
4
Enter your age
48

Governement Schemes available under this section are:
1.Pradhan Mantri Jan Dhan Yojna (PMJDY)
3.Rajiv Gandhi Jeevandayee Arogya Yojana (RGJAY)
4.Pradhan Mantri Jeevan Jyoti Bima Yojna (PMJJBY)
5.Rashtriya Swasthya Bima Yojana (RSBY)
6.National Social Assistance scheme
7.Pradhan Mantri Mudra Yojana (PMMY)
8.PM-KISAN (Pradhan Mantri Kisan Samman Nidhi) Scheme
9.Pradhan Mantri Kisan Pension Yojana
10.Pradhan Mantri Shram Yogi Maan-dhan (PM-SYM)
11.Varishtha Pension Bima Yojana (VPBY)
12.PRIME MINISTER'S SPECIAL SCHOLARSHIP SCHEME - PMSSS
***THANK YOU FOR VISTING***
```

Fig 6.5.2 Government schemes.

## 6.6 Displaying

In the below image user selected option two for displaying all the details stored in the minicensus so it shows the every house details by making columns and rows.

```
1:Login
2:Display
3:Search
4:Government schemes
5:Exit
select your choice:
2
```

village	House no	Names	Age	Gender
Honnali.	101	Rakesh.M	46	male
Honnali.	101	Savitha.R	41	female
Honnali.	101	Srujan.RM	21	male
Honnali.	101	Pooja.Rm	18	female

```

      *           *           *
    _ _         _ _         _ _
```

```

Rampura.      201      Maheshappa.K      79      male
Rampura.      201      Umesh.KM      46      male
Rampura.      201      Sumithra.M      41      female
Rampura.      201      Harshitha.KM      25      female
Rampura.      201      Karthik.KM      21      male
      *
      - -
Harihara.      301      Praveen.HC      39      male
Harihara.      301      Padmavathi.H      35      female
Harihara.      301      Kushi.HP      14      female
Harihara.      301      Harsh.HP      10      male
      *
      - -
Rampura.      202      Sharadhadevi.KP      73      female
Rampura.      202      Girish.KP      51      male
Rampura.      202      Shobha.K      47      female
Rampura.      202      Nandini.KG      28      female
Rampura.      202      Abhishek.KG      25      male
      *
      - -
Kulagate.      401      Halesh.AK      43      male
Kulagate.      401      Kavitha.P      39      female
Kulagate.      401      Sidharth.AH      19      male
Kulagate.      401      Pavan.AH      16      male
      *
      - -
*Total number of houses:5
*Total number of peoples:22
*Number of males:12
*Number of females:10
*Number of peoples below 18 years:3
*Number of peoples between (18-65) years:17
*Number of peoples above 65 years:2

```

Fig 6.6.1 Displaying.



### CHAPTER 7

## CONCLUSION

This program makes the user simpler to collect the census of village. The entire house details stored under single house number , this is one benefit to user to simply search and locates his required house details. This program deals with mainly four operations of create details, deleting them, modifying, and searching according the user's choice. Each operation is made as an individual function and so control enters to different structures and all the data added, modified or deleted is going to be store and there is a government schemes option is available so village peoples can get know about all the schemes by entering their age.

## REFERENCES

- Java™:The Complete Reference, Seventh Edition(Herbert Schildt).
- www.javatpoint.com (<https://www.javatpoint.com/java-oops-concepts>).
- www.tutorialspoint.com ([https://www.tutorialspoint.com/java/java\\_overview.htm#:](https://www.tutorialspoint.com/java/java_overview.htm#:))
- www.geeksforgeeks.org (<https://www.geeksforgeeks.org/java/>).
- www.w3schools.com (<https://www.w3schools.com/java/>).