INTERNSHIP REPORT

A Report for Internship Programme

Submitted by

Premanshu Sharma

Stream: Computer Science and Engineering

Year: 4th

Section: B

Class Roll No.: 098

University Roll No.: 430120010098

University Registration No.: 201270100110081

College: Narula Institute of Technology

At

Persistent Systems

CONTENTS

Date

Signature

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to the team at **Persistent Systems** for providing me with the opportunity to undertake my internship at their esteemed organization.

I extend my heartfelt thanks to **Mr. Sandeep Kalra, The CEO of Persistent Systems**, whose vision and leadership have created an environment that fosters learning and innovation. The guidance and insights I received from **Mr. Sandeep Kalra** significantly contributed to my professional development during this internship.

I am deeply appreciative of the Human Resources department at Persistent Systems for their support and assistance throughout my internship. Their commitment to creating a positive and inclusive workplace has made a lasting impression on my experience.

Special thanks to Ms. Kalpana Kudlingar, Head of Early Careers and Campus Relations, for her unwavering support and mentorship. Her dedication to nurturing talent and providing valuable guidance has been instrumental in shaping my internship journey.

I am grateful for the invaluable experiences, knowledge, and skills I have gained during my time at Persistent Systems. This internship has been a transformative learning experience, and I look forward to applying these insights as I continue to grow in my career.

Thank you to everyone at Persistent Systems for making my internship a memorable and enriching experience.

PREMANSHU SHARMA

Internship Company Background Information: Persistent Systems

Persistent Systems is not only recognized for its innovative technology solutions but is also known for its robust internship programs and commitment to nurturing talent. The company offers dynamic internship opportunities, providing students with hands-on experience in cutting-edge projects. Through mentorship and exposure to real-world challenges, interns gain valuable skills and insights.

Furthermore, **Persistent Systems** has a strong track record of absorbing talented individuals from its internship programs into full-time positions. The company's emphasis on early career development, guided by leaders like **Ms. Kalpana Kudlingar, Head of Early Careers and Campus Relations**, underscores its dedication to cultivating a skilled workforce for the future. This commitment to talent development makes Persistent Systems an attractive destination for those seeking both meaningful internships and long-term career opportunities.

URL: PERSISTENT SYSTEMS

Address: 402, Senapati Bapat Rd, Shivaji Co-operative Housing Society, Bhageerath, Gokhalenagar.

Email ID: corpsec@persistent.com

Internship Position

Role: Student Intern

As a student intern at Persistent Systems, I undertook a dynamic role focused on individual exploration and mastery of Core Java concepts, with a particular emphasis on Java applet development and the implementation of threads. This experience provided me with a comprehensive understanding of both fundamental and advanced Java programming principles.

Responsibilities:

1.**Individual Learning Journey:** Independently delved into Core Java, comprehensively understanding, and implementing the five types of Object-Oriented Programming (OOPs) concepts.

2.**Applet Development:** Acquired hands-on experience in creating dynamic and interactive Java applets, applying learned concepts to real-world projects.

3.**Thread Implementation:** Implemented and explored the intricacies of multithreading in Java, gaining insights into concurrent programming and synchronization.

4.**Self-Directed Research:** Conducted independent research to stay abreast of the latest advancements in Java technology, with a specific focus on threading concepts.

Focus on Core Java, Applet, and Thread Implementation

1.1 Core Java Concepts:

1.1.1 In-depth Exploration:

Engaged in a comprehensive study of Core Java, covering foundational concepts such as variables, data types, control structures, and methods.

Delved into the nuances of Object-Oriented Programming (OOP) principles, including encapsulation, inheritance, polymorphism, abstraction, and interfaces.

1.1.2 Hands-on Practice:

Applied theoretical knowledge through practical exercises and coding challenges, solidifying understanding and proficiency in Core Java fundamentals.

1.2 Java Applet Development:

1.2.1 Understanding Applet Architecture:

Gained insights into the architecture of Java applets, understanding their lifecycle, and the integration of applets within web applications.

1.2.2 Interactive UI Design:

Developed skills in creating interactive user interfaces using Java applets, incorporating graphical elements and event-driven programming for a dynamic user experience.

1.2.3 Real-world Application:

Applied applet development skills to real-world projects, demonstrating the ability to translate conceptual understanding into practical solutions.

1.3 Thread Implementation:

1.3.1 Conceptual Grasp:

Studied the principles of multithreading in Java, understanding the benefits and challenges associated with concurrent programming.

1.3.2 Hands-on Implementation:

Actively implemented threads in Java, exploring the creation, synchronization, and coordination of multiple threads to achieve parallel execution.

1.3.3 Troubleshooting and Optimization:

Addressed challenges related to thread safety, synchronization, and race conditions, gaining practical experience in troubleshooting and optimizing multithreaded programs.

1.4 Independent Research:

1.4.1 Continuous Learning:

Conducted independent research to stay updated on the latest trends and best practices in Core Java, applet development, and multithreading.

1.4.2 Application of New Concepts:

Incorporated new concepts and advancements into personal projects, fostering a culture of continuous improvement and staying at the forefront of Java technologies.

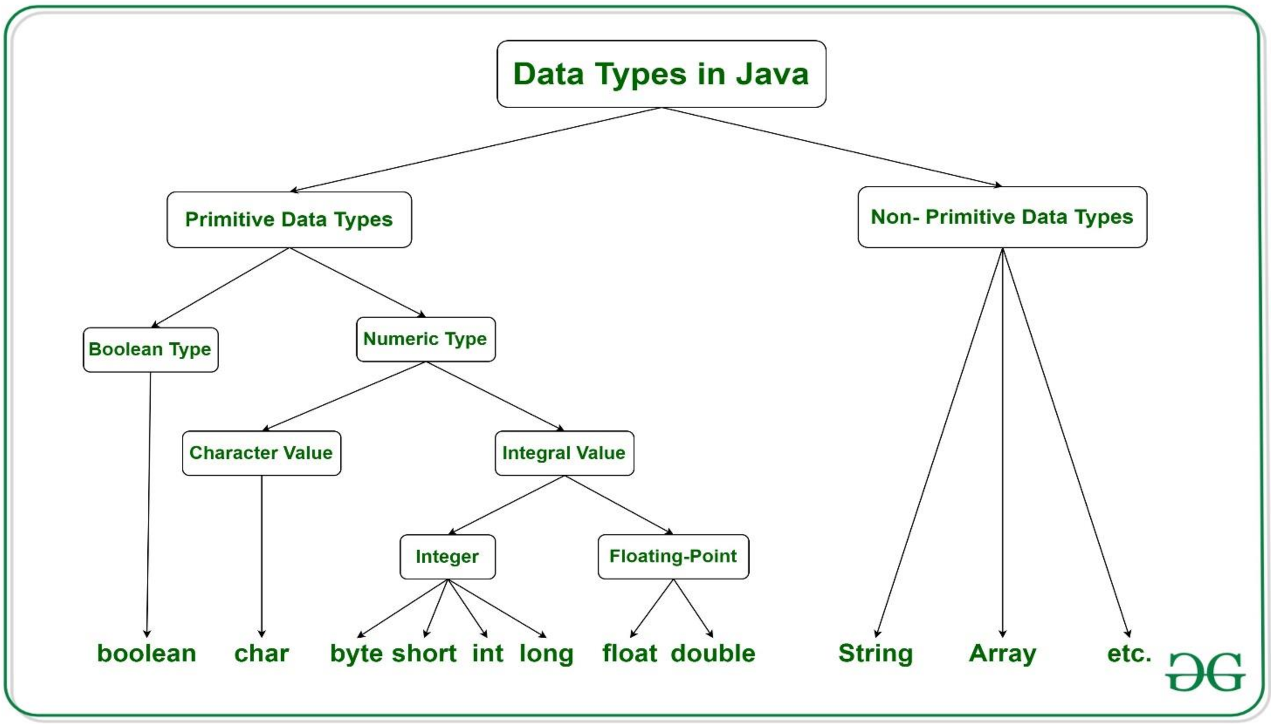
1.5 Challenges and Solutions:

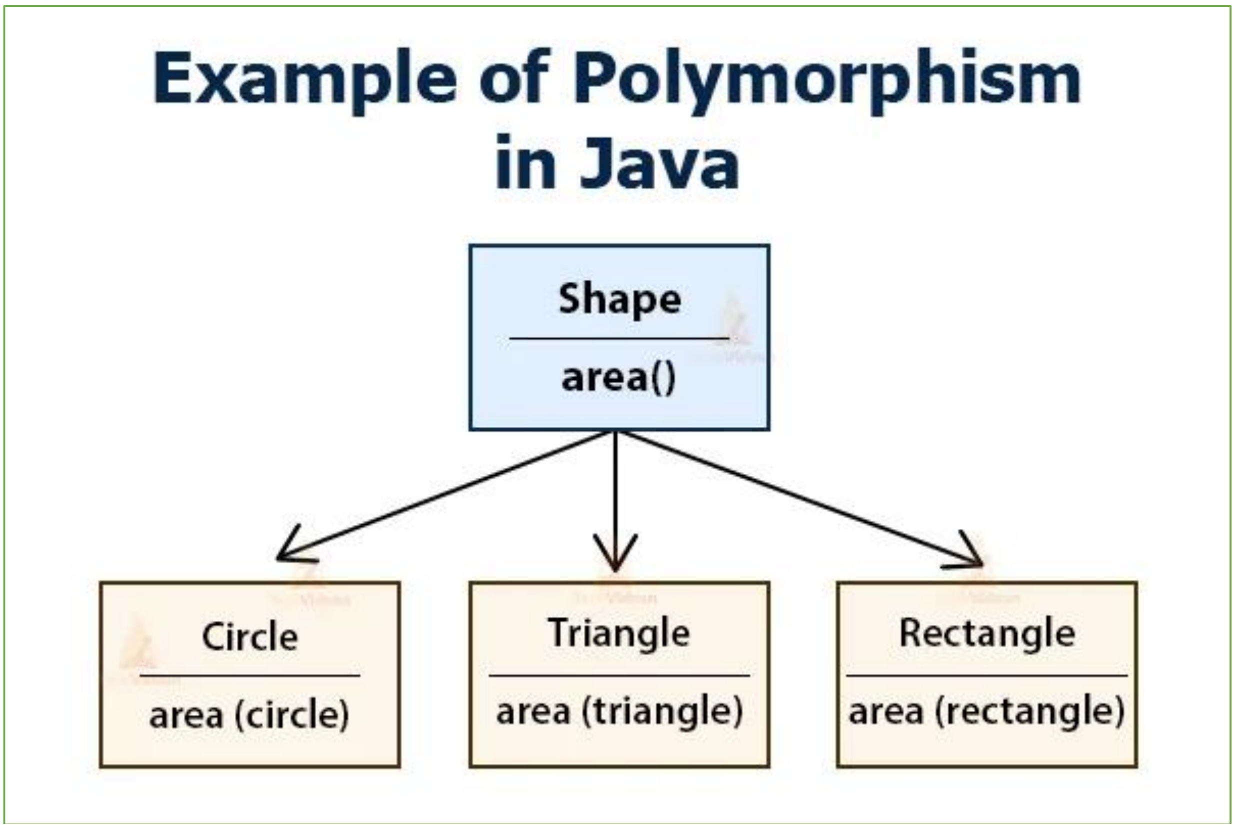
1.5.1 Adaptive Problem-Solving:

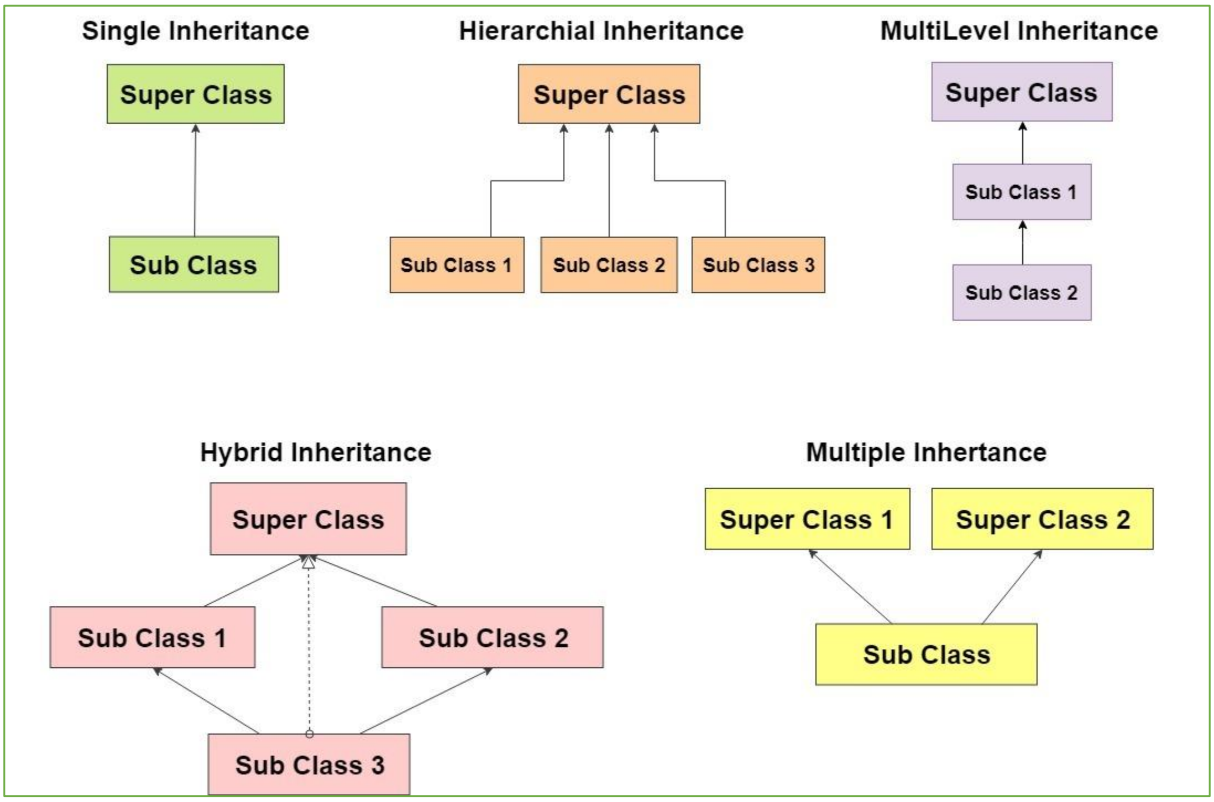
Encountered challenges in the implementation of Core Java, applet development, and threading, demonstrating adaptability and problem-solving skills.

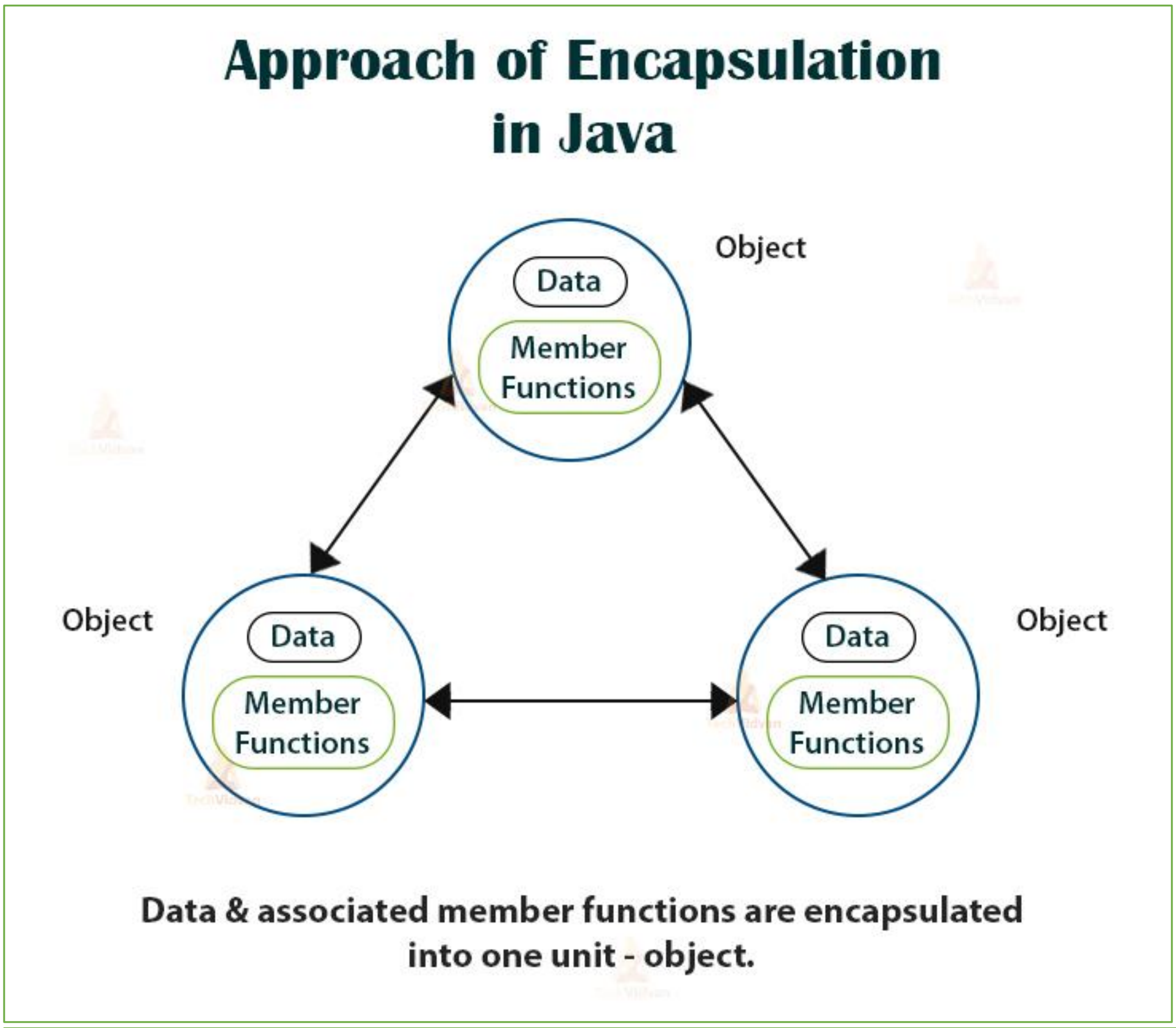
1.5.2 Reflective Learning:

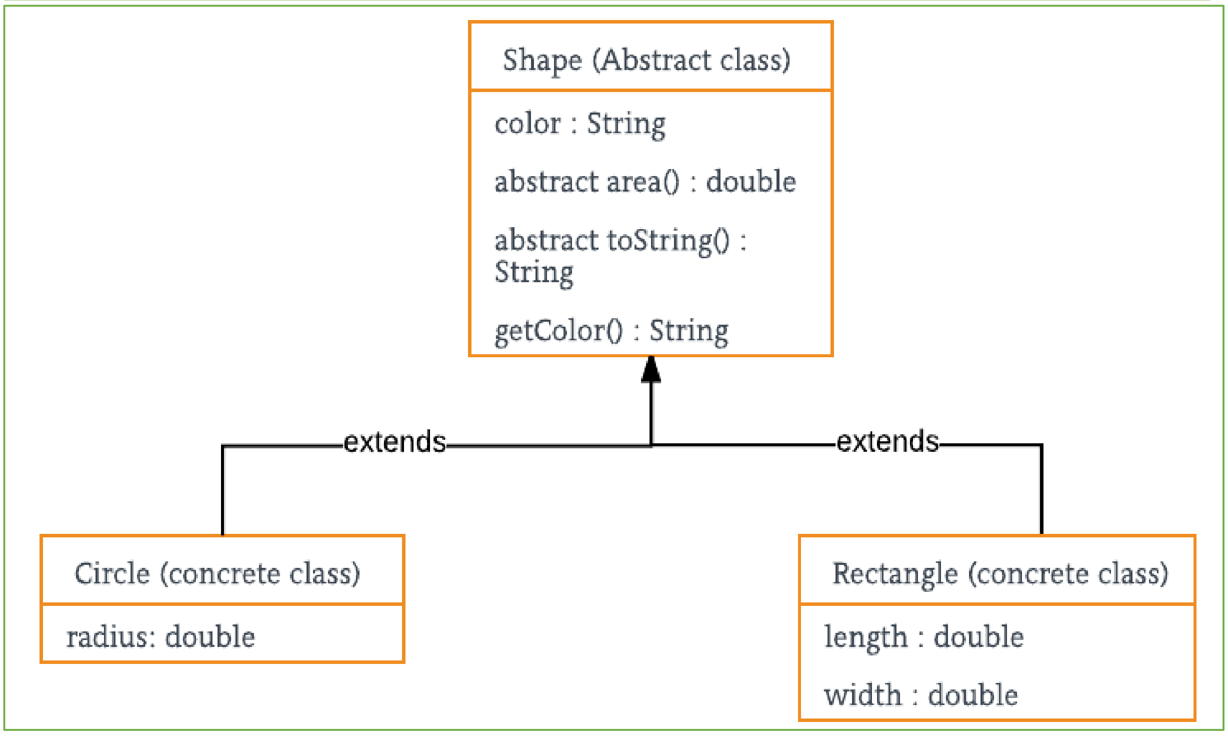
Used challenges as opportunities for learning, reflecting on solutions to enhance understanding and skill refinement.

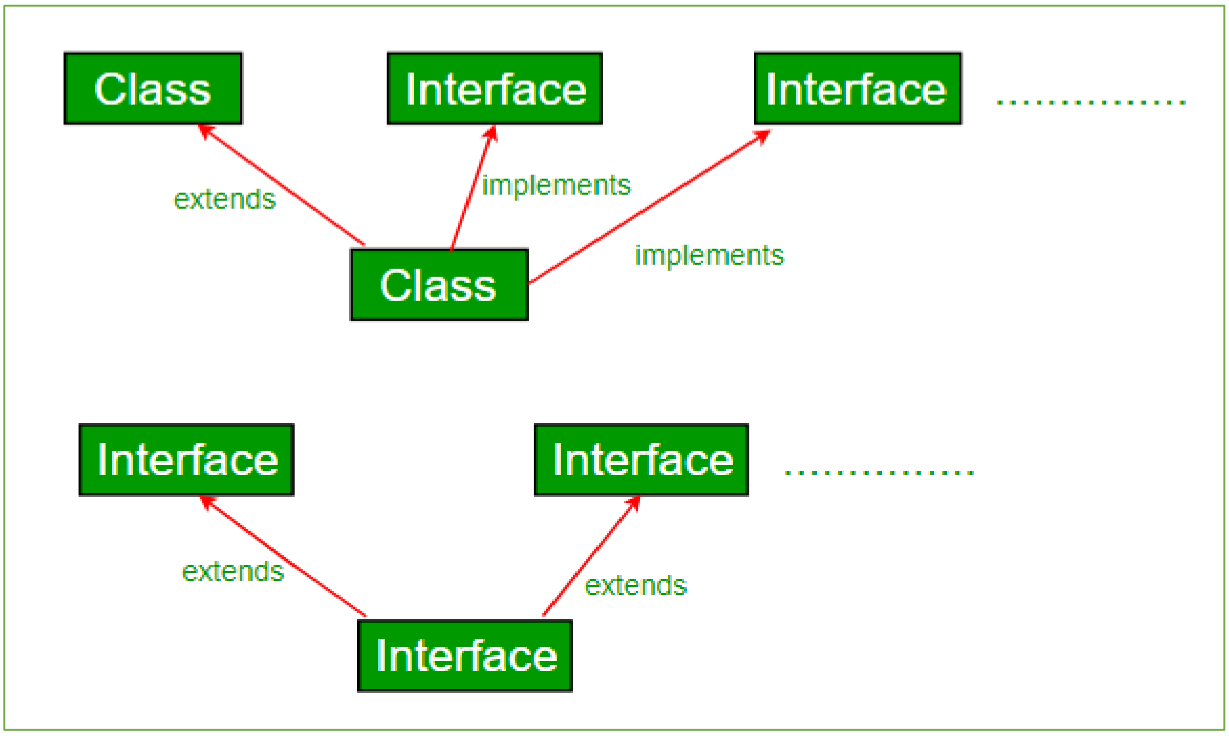

Data Types in Java
Primitive Data Types
Non- Primitive Data Types
Boolean Type
Numeric Type
Character Value
Integral Value
Integer
Floating-Point
boolean
char
byte short int long
float double
String
Array
etc.


Example of Polymorphism in Java
Shape
area()
Circle
area (circle)
Triangle
area (triangle)
Rectangle
area (rectangle)


Single Inheritance
Super Class
Sub Class
Hierarchial Inheritance
Super Class
Sub Class 1
Sub Class 2
Sub Class 3
MultiLevel Inheritance
Super Class
Sub Class 1
Sub Class 2
Hybrid Inheritance
Super Class
Sub Class 1
Sub Class 2
Sub Class 3
Multiple Inhertance
Super Class 1
Super Class 2
Sub Class

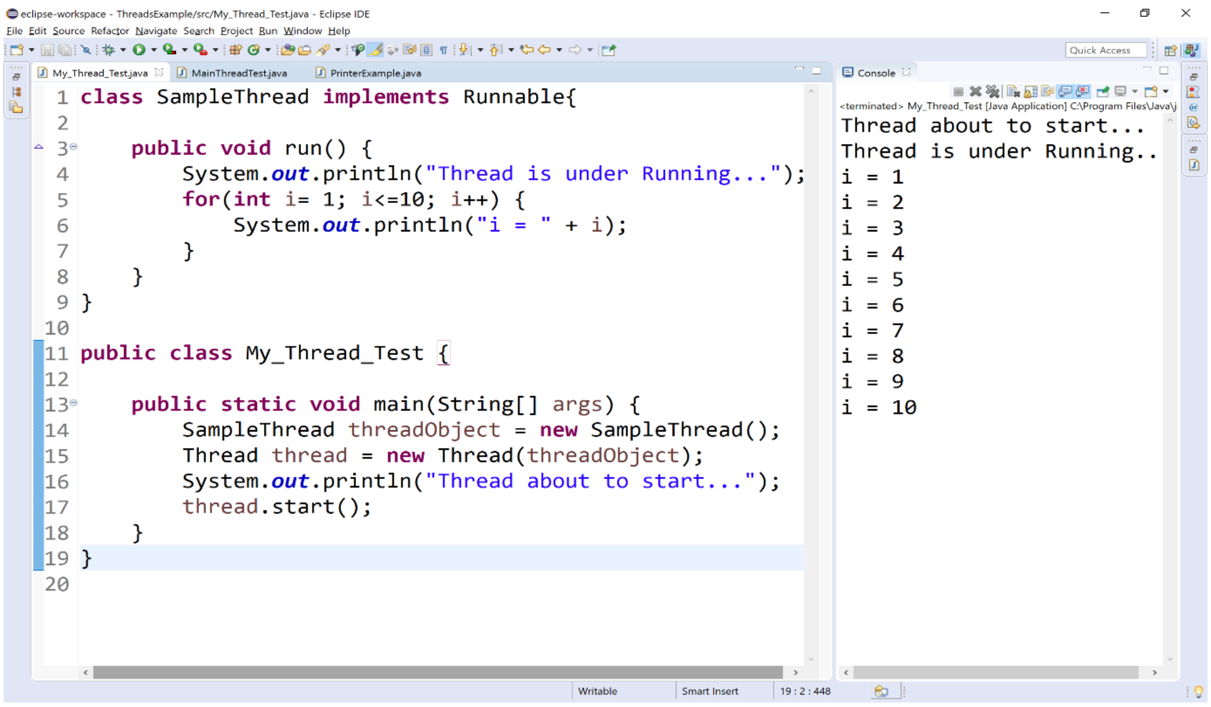

Approach of Encapsulation in Java
Object
Data
Member Functions
Object
Data
Member Functions
Object
Data
Member Functions
Data & associated member functions are encapsulated into one unit - object.


Shape (Abstract class)
color : String
abstract area() : double
abstract toString() : String
getColor() : String
extends
extends
Circle (concrete class)
radius: double
Rectangle (concrete class)
length : double
width : double


Class
Interface
Interface
....
extends
implements
implements
Class
Interface
Interface
....
extends
extends
Interface

CODE SNIPPETS

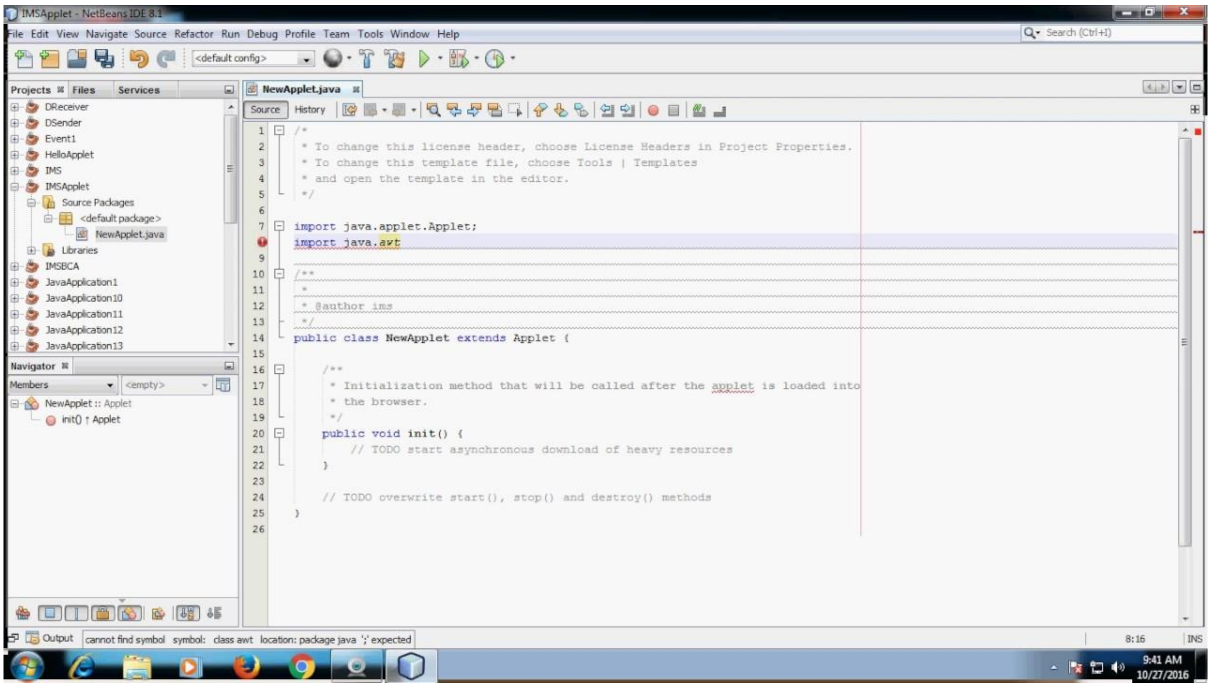
- **Thread**



```java
class SampleThread implements Runnable{

    public void run() {
        System.out.println("Thread is under Running...");
        for(int i= 1; i<=10; i++) {
            System.out.println("i = " + i);
        }
    }
}

public class My_Thread_Test {

    public static void main(String[] args) {
        SampleThread threadObject = new SampleThread();
        Thread thread = new Thread(threadObject);
        System.out.println("Thread about to start...");
        thread.start();
    }
}
```

```
<terminated> My_Thread_Test [Java Application] C:\Program Files\Java\j
Thread about to start...
Thread is under Running..
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
```

- **Applet**



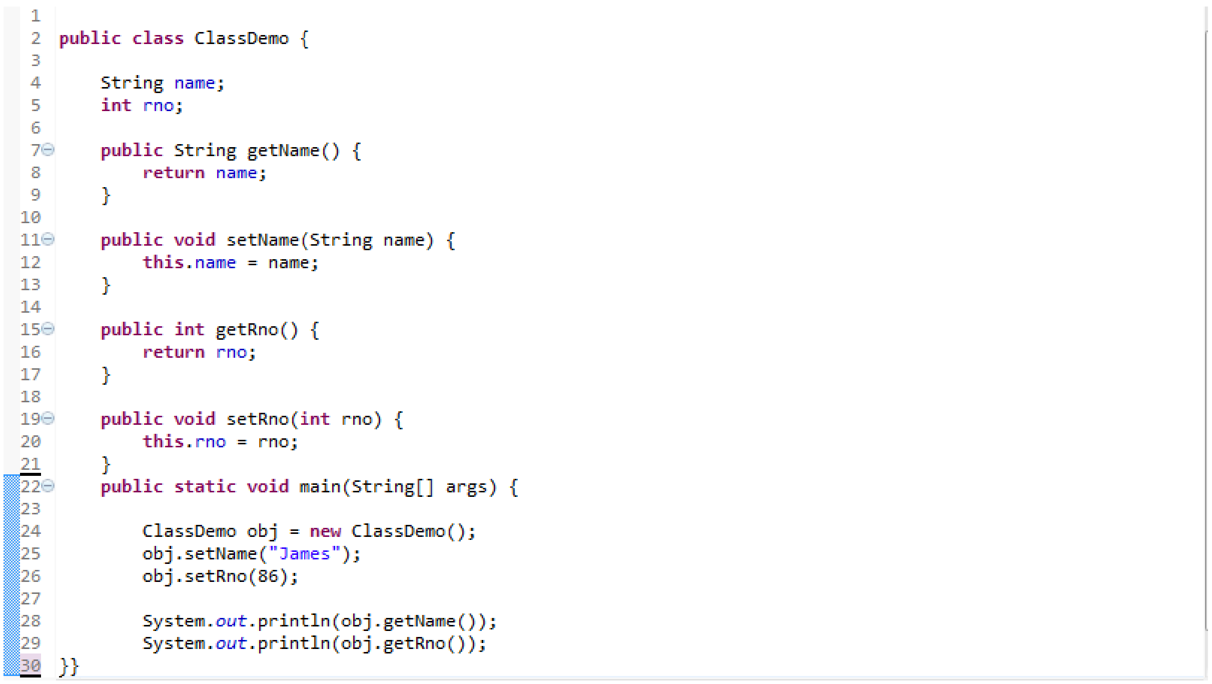
```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

import java.applet.Applet;
import java.awt

/**
 *
 * @author ims
 */
public class NewApplet extends Applet {

    /**
     * Initialization method that will be called after the applet is loaded into
     * the browser.
     */
    public void init() {
        // TODO start asynchronous download of heavy resources
    }

    // TODO overwrite start(), stop() and destroy() methods
}
```

```
cannot find symbol  symbol:  class awt  location: package java  ';' expected
```

- Inheritance

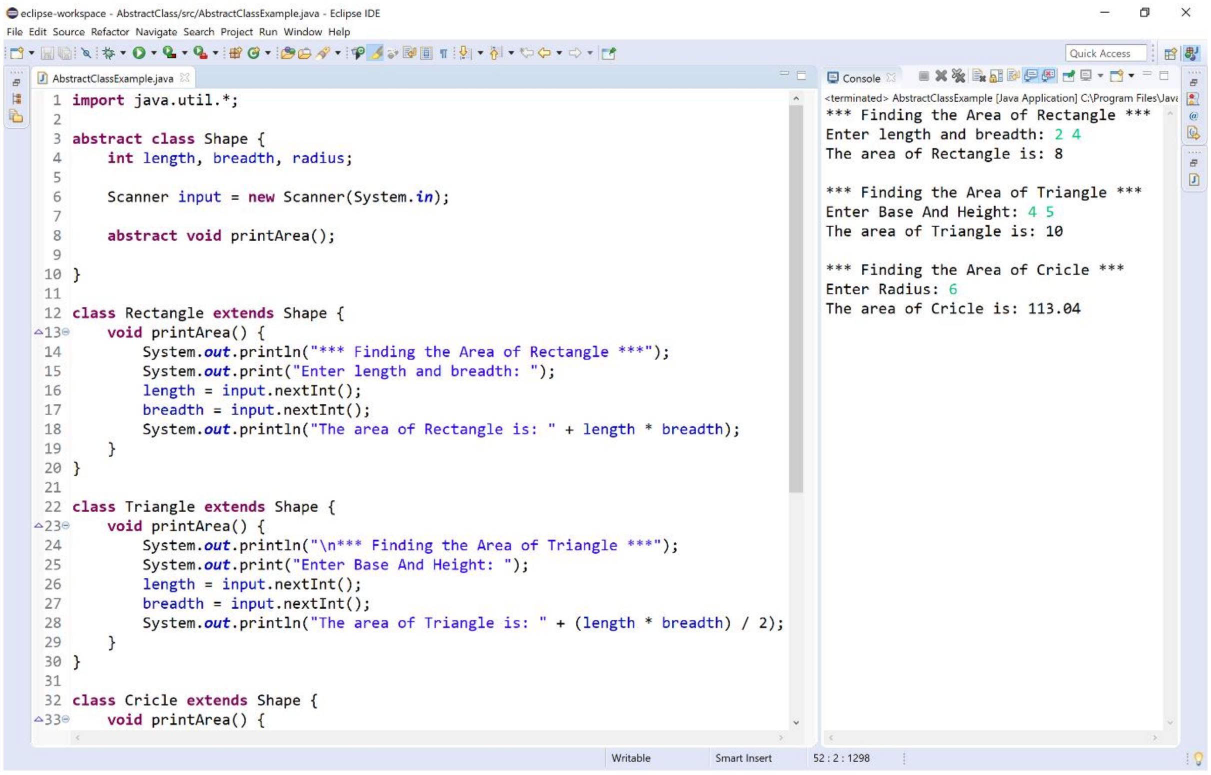
```
class Animal{
String color="white";
}
class Dog extends Animal{
String color="black";
void printColor(){
System.out.println(color);//prints color of Dog class
System.out.println(super.color);//prints color of Animal class
}
}
class TestSuper1{
public static void main(String args[]){
Dog d=new Dog();
d.printColor();
}}
```

- Encapsulation



```

public class ClassDemo {

    String name;
    int rno;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getRno() {
        return rno;
    }

    public void setRno(int rno) {
        this.rno = rno;
    }
    public static void main(String[] args) {

        ClassDemo obj = new ClassDemo();
        obj.setName("James");
        obj.setRno(86);

        System.out.println(obj.getName());
        System.out.println(obj.getRno());
}}
```

- Data Abstraction



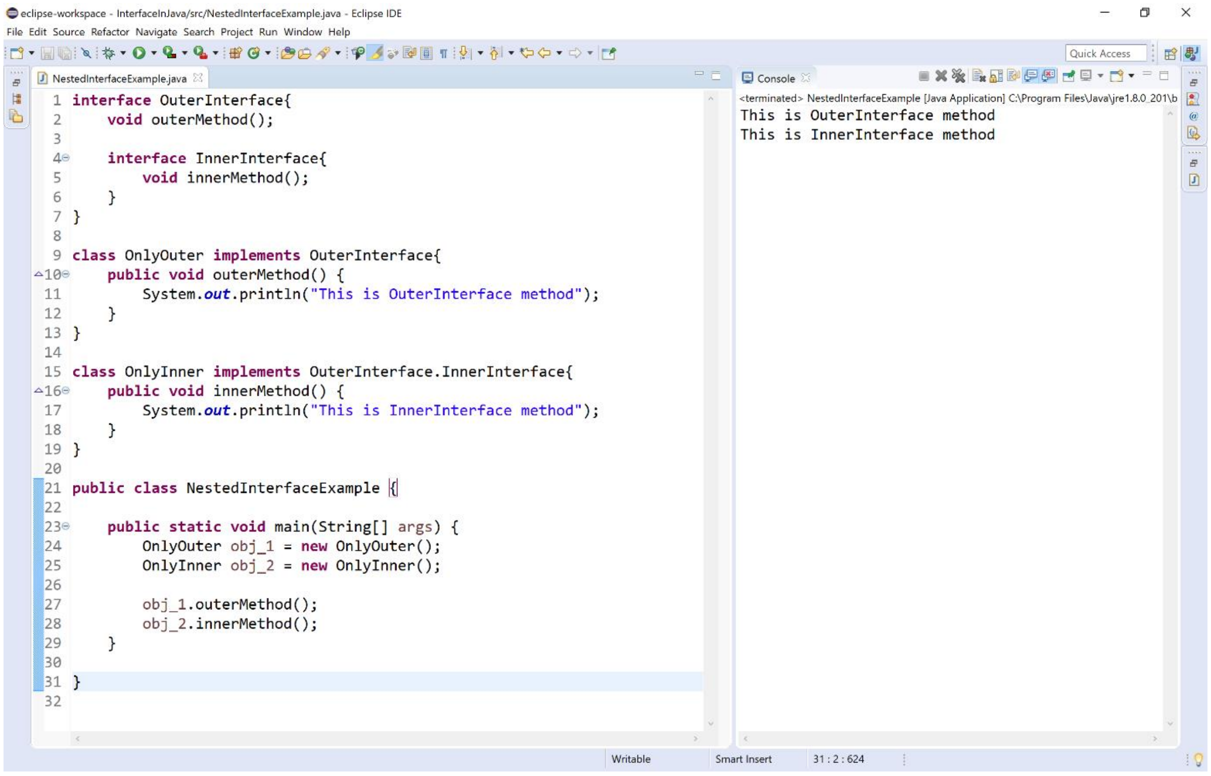
eclipse-workspace - AbstractClass/src/AbstractClassExample.java - Eclipse IDE

```java
import java.util.*;

abstract class Shape {
    int length, breadth, radius;

    Scanner input = new Scanner(System.in);

    abstract void printArea();

}

class Rectangle extends Shape {
    void printArea() {
        System.out.println("*** Finding the Area of Rectangle ***");
        System.out.print("Enter length and breadth: ");
        length = input.nextInt();
        breadth = input.nextInt();
        System.out.println("The area of Rectangle is: " + length * breadth);
    }
}

class Triangle extends Shape {
    void printArea() {
        System.out.println("\n*** Finding the Area of Triangle ***");
        System.out.print("Enter Base And Height: ");
        length = input.nextInt();
        breadth = input.nextInt();
        System.out.println("The area of Triangle is: " + (length * breadth) / 2);
    }
}

class Cricle extends Shape {
    void printArea() {
```

```
<terminated> AbstractClassExample [Java Application] C:\Program Files\Java
*** Finding the Area of Rectangle ***
Enter length and breadth: 2 4
The area of Rectangle is: 8

*** Finding the Area of Triangle ***
Enter Base And Height: 4 5
The area of Triangle is: 10

*** Finding the Area of Cricle ***
Enter Radius: 6
The area of Cricle is: 113.04
```

Writable Smart Insert 52 : 2 : 1298

- Interface



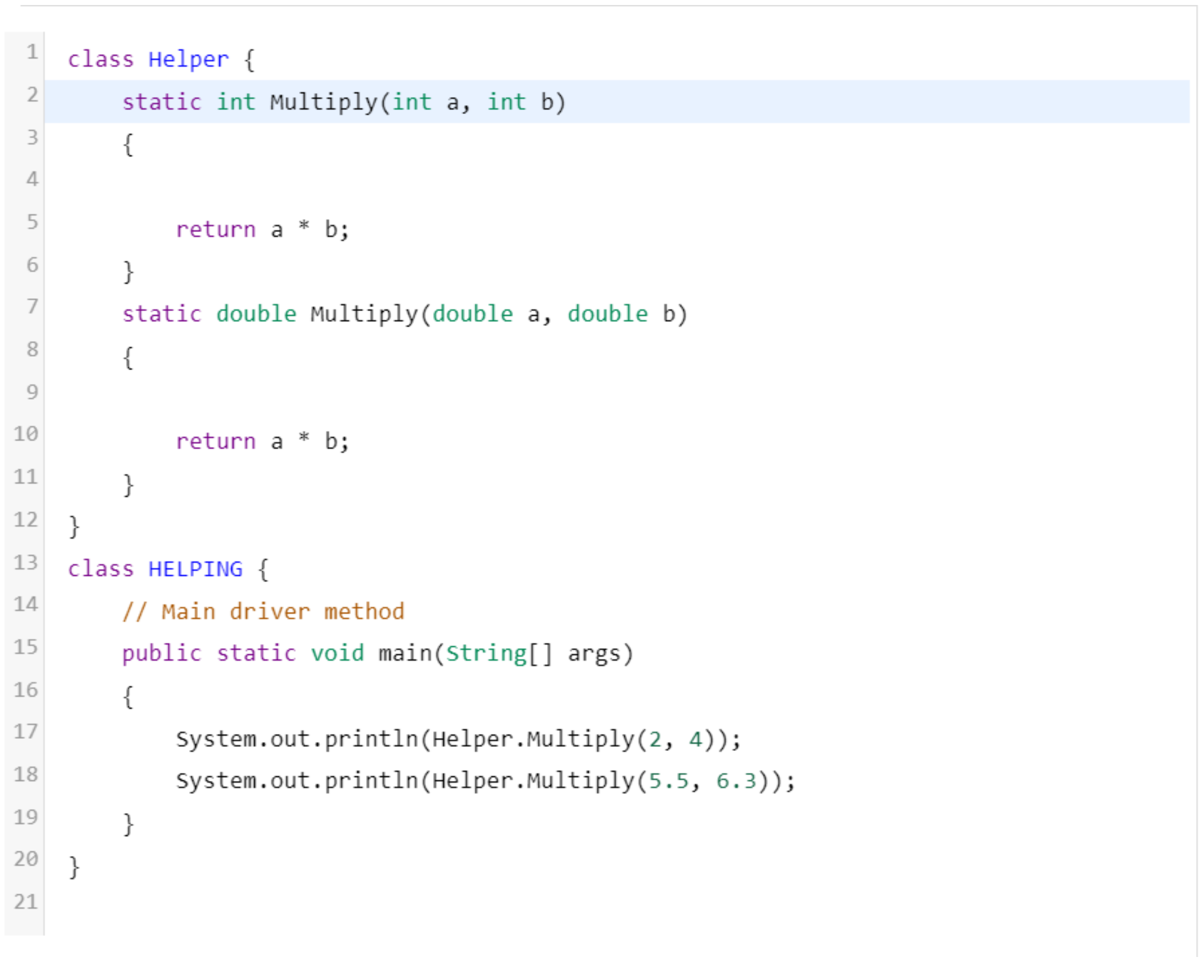
eclipse-workspace - InterfaceInJava/src/NestedInterfaceExample.java - Eclipse IDE

```java
interface OuterInterface{
    void outerMethod();

    interface InnerInterface{
        void innerMethod();
    }
}

class OnlyOuter implements OuterInterface{
    public void outerMethod() {
        System.out.println("This is OuterInterface method");
    }
}

class OnlyInner implements OuterInterface.InnerInterface{
    public void innerMethod() {
        System.out.println("This is InnerInterface method");
    }
}
public class NestedInterfaceExample {

    public static void main(String[] args) {
        OnlyOuter obj_1 = new OnlyOuter();
        OnlyInner obj_2 = new OnlyInner();

        obj_1.outerMethod();
        obj_2.innerMethod();
    }

}
```

```
<terminated> NestedInterfaceExample [Java Application] C:\Program Files\Java\jre1.8.0_201\b
This is OuterInterface method
This is InnerInterface method
```

Writable Smart Insert 31 : 2 : 624

- Polymorphism



```java
class Helper {
    static int Multiply(int a, int b)
    {

        return a * b;
    }
    static double Multiply(double a, double b)
    {

        return a * b;
    }
}
class HELPING {
    // Main driver method
    public static void main(String[] args)
    {
        System.out.println(Helper.Multiply(2, 4));
        System.out.println(Helper.Multiply(5.5, 6.3));
    }
}

```

Technical Skills Acquired

1.1 Programming Proficiency:

- **Deepened Understanding:**
  - Enhanced my understanding of Core Java, solidifying foundational programming concepts such as variables, data types, control structures, and methods.
- **Application of Concepts:**
  - Applied theoretical knowledge in practical scenarios, gaining proficiency in creating efficient and well-structured Java code.

1.2 Hands-on Applet Experience:

- **Project Application:**
  - Applied learned concepts to Java applet development, gaining hands-on experience in creating interactive and visually engaging user interfaces.
- **Real-world Application:**
  - Worked on real-world projects incorporating Java applets, addressing challenges associated with integration and user experience.

1.3 Challenges and Problem-Solving:

- **Overview of Challenges:**
  - Encountered challenges in individual learning, applet development, and thread implementation, ranging from conceptual difficulties to practical coding issues.
- **Adaptive Problem-Solving:**
  - Employed adaptive problem-solving strategies to overcome challenges, fostering resilience and a proactive approach to technical issues.

Self-Directed Learning

1.1 Emphasis on Individual Learning and Research:

- **Curiosity-Driven Exploration:**
    - Emphasized individual learning by exploring advanced Java concepts beyond the internship curriculum, driven by curiosity and a thirst for knowledge.

- **Independent Research:**
    - Conducted independent research on online platforms, official documentation, and technical publications to deepen understanding and broaden skill sets.

1.2 Tools and Resources Utilized for Self-Training:

- **Online Learning Platforms:**
    - Leveraged online learning platforms to access tutorials, courses, and coding exercises to reinforce theoretical knowledge with practical application.

- **Community Engagement:**
    - Participated in online coding communities and forums to seek advice, share experiences, and engage in collaborative problem-solving.

Professional Development

1.1 Workshops, Training, or Self-Directed Learning:

- **Interactive Workshops:**
    - Participated in interactive workshops facilitated by industry experts, focusing on advanced Java topics, applet development, and threading.

- **Self-Directed Learning Paths:**
    - Engaged in self-directed learning paths, exploring specialized areas of interest and gaining practical insights beyond the formal curriculum.

1.2 Certifications or Skill Badges Earned:

- **Recognition of Achievement:**
    - Earned certifications or skill badges as a testament to the acquired proficiency in Core Java, applet development, and thread implementation.

- **Continuous Learning:**
    - Demonstrated a commitment to continuous learning by pursuing and obtaining certifications that validate expertise in specific technical domains.

Self-Directed Research

1.1 Continuous Learning:

- **Proactive Exploration:**
    - Demonstrated a proactive approach to learning by conducting self-directed research on advanced Java topics beyond the internship curriculum.

- **Online Resources and Documentation:**
    - Utilized online resources, official Java documentation, and reputable technical publications to delve into intricate aspects of Core Java, applet development, and threading.

1.2 Application of New Concepts:

- **Personal Projects:**
    - Incorporated new concepts and advancements gained from research into personal coding projects, showcasing the practical application of the latest Java technologies.

- **Experimental Coding:**
    - Engaged in experimental coding exercises to test and solidify understanding, fostering a culture of curiosity and innovation.

Learning Structure

1.1 Focused Learning Environment:

- **Curriculum Emphasis:**
  - Engaged in a structured curriculum designed to enhance core Java competencies, with a specialized focus on applet development and thread implementation.

- **Interactive Learning Modules:**
  - Participated in interactive learning modules that facilitated a deeper understanding of complex concepts through practical exercises and real-world examples.

1.2 Guidance and Mentorship:

- **Industry Experts:**
  - Received guidance and mentorship from industry experts within Persistent Systems, leveraging their insights to enhance theoretical understanding and practical application.

- **Ms. Kalpana Kudlingar's Role:**
  - Ms. Kalpana Kudlingar, Head of Early Careers and Campus Relations, played a crucial role in providing guidance and support throughout the learning process, contributing to a positive and enriching internship experience.

Mentorship and Guidance

1.1 Role of Mentors in Providing Guidance:

- **Guided Learning Path:**
  - Mentors played a crucial role in shaping the learning path, providing insights into the nuances of Core Java, applet development, and thread implementation.

- **Clarification of Concepts:**
  - Mentors offered valuable clarifications and explanations, ensuring a deeper understanding of complex topics and practical applications.

1.2 Key Takeaways from Mentorship:

- **Practical Industry Insights:**
  - Gained practical insights into industry best practices and real-world applications through mentorship, bridging the gap between academic knowledge and professional implementation.

- **Career Guidance:**
  - Received guidance on potential career paths, industry trends, and areas for specialization within the realm of Java development.

Achievements and Milestones:

1.1 Personal Achievements:

- **Mastery of Core Java:**
  - Achieved mastery in Core Java, demonstrating proficiency in creating efficient and scalable Java applications.

- **Successful Applet Development:**
  - Successfully designed and implemented Java applets in real-world scenarios, showcasing an ability to create engaging and interactive user interfaces.

- **Thread Implementation Proficiency:**
  - Demonstrated expertise in the implementation of threads, contributing to a deeper understanding of concurrent programming.

1.2 Recognition or Self-Evaluations:

- **Positive Feedback:**
  - Received positive feedback from mentors, peers, and self-evaluations, acknowledging achievements in individual learning, applet development, and threading.

- **Acknowledgment of Growth:**
  - Self-reflected on personal and professional growth, recognizing areas of improvement and celebrating milestones achieved during the internship.

Reflection on Learning Objectives

1.1 Assessing Achievements in Relation to Initial Learning Goals:

- **Goal Attainment:**
  - Evaluated the extent to which initial learning goals were achieved, considering advancements in Core Java, applet development, and thread implementation.
- **Self-Reflection:**
  - Engaged in reflective practices to assess personal and technical growth, aligning achievements with the initial objectives set at the beginning of the internship.

1.2 Areas for Future Improvement:

- **Identifying Growth Opportunities:**
  - Identified areas for future improvement, outlining a roadmap for continuous learning and skill enhancement beyond the internship.
- **Feedback Utilization:**
  - Incorporated feedback received during the internship as a guide for future improvement, ensuring a proactive approach to ongoing development.

Conclusion:

1.1 Summary of Key Learnings:

- **Holistic Java Mastery:**
    - Summarized key learnings, showcasing a holistic mastery of Core Java, applet development, and thread implementation.

- **Skill Enhancement:**
    - Highlighted the enhancement of technical skills, problem-solving abilities, and a comprehensive understanding of Java programming.

1.2 Gratitude and Acknowledgments:

- **Expression of Gratitude:**
    - Expressed gratitude to mentors, colleagues, and the entire Persistent Systems team for their support, guidance, and enriching learning experiences.

- **Acknowledgment of Impact:**
    - Acknowledged the positive impact of the internship on personal and professional growth, expressing appreciation for the opportunities provided.

Appendices:

- **Supporting Documents:**
    - Included relevant document such as certificate that complement the internship report.
- **Code Samples:**
    - Provided snippets of code developed during the internship, showcasing practical application and coding proficiency.
- **Visuals:**
    - Included visuals, such as diagrams or charts, that enhance the understanding of specific achievements or project outcomes.