MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

MICRO-PROJECT REPORT ON

“Bank Management System”

Diploma in Computer Engineering

Submitted By

Name: Pruthesh Neminath Upadhye

Roll No:-2073

Subject: Java Programming

Guided By: Miss S.I.Mulla

ASHOKRAO MANE POLYTECHNIC,VATHAR

Department of Computer Engineering

A.Y.2023-24

DECLARATION

I hereby declare that the Micro-Project Report entitled " **JAVA PROGRAMMING ”** with Module **“Bank Management System”** is an authentic record of my own work as requirements of Micro Project during the period from **/ /2024** to / **/ 2024** in partial fulfillment of **Second Year** in **Computer Engineering, Ashokrao Mane Polytechnic, Vathar** under the guidance of **Miss.S.I.Mulla**

Signature of Student

Pruthesh Neminath Upadhye

Roll no: 2073

Date: / / 2024

Place: Vathar, Kolhapur

ACKNOWLEDGMENT

The successful presentation of the **Java Programming Micro Project** would be in complete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I would like to extend my gratitude to **Mr S.A.Lakade** Kolhapur, for providing all the facilities to present this Micro-Project.

I would like to thank our Project Guide, **Miss S.I.Mulla**, Department of **Computer Engineering, Ashokrao Mane Polytechnic,Vathar**, for their constant guidance and in puts.

Sincerely,

Pruthesh Neminath Upadhye

Place: Vathar, Kolhapur.

ABSTRACT

This project aims to develop a robust Bank Management System using Java programming language. The system facilitates efficient management of various banking operations such as customer account management, transaction processing, loan management, and employee management. Key features include a user-friendly interface for customers and bank staff, secure authentication mechanisms, real-time transaction tracking, and comprehensive reporting functionalities. Through object-oriented design principles and modular architecture, the system ensures scalability and maintainability. Additionally, the implementation adheres to industry standards for data security and privacy, ensuring the confidentiality and integrity of customer information. Overall, this project offers a reliable solution to streamline banking operations and enhance customer satisfaction.

INDEX

1. Introduction to Java Programming

1.1 What is Java?

Java is a high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle Corporation). It was designed to be platform-independent, meaning that Java programs can run on any device that has a Java Virtual Machine (JVM) installed, regardless of the underlying hardware and operating system. Java is renowned for its simplicity, portability, and robustness, making it a popular choice for building a wide range of applications, including web applications, mobile apps, desktop software, enterprise systems, and more. It features automatic memory management (garbage collection), strong type checking, and extensive libraries, making it suitable for both beginners and experienced developers. Additionally, Java's "write once, run anywhere" philosophy has contributed to its widespread adoption and versatility in the software development industry.

❖ **Key components of a JAVA include:**

1. Java Development Kit (JDK)
2. Java Virtual Machine (JVM)
3. Java Standard Edition (Java SE)
4. Java Enterprise Edition (Java EE)
5. Java Micro Edition (Java ME)
6. JavaFX

2. Introduction to Object Oriented

2.1 What is Object Oriented Language ?

An object-oriented programming (OOP) language is a programming paradigm based on the concept of "objects," which can contain data in the form of fields or attributes, and code in the form of procedures or methods. OOP languages enable developers to model real-world entities as objects and define their interactions through encapsulation, inheritance, and polymorphism. Encapsulation refers to the bundling of data and methods within an object, providing control over access and modification. Inheritance allows objects to inherit properties and behaviors from parent classes, facilitating code reuse and hierarchy. Polymorphism enables objects of different types to be treated uniformly, allowing for flexibility and extensibility in the code. OOP languages promote modularity, reusability, and maintainability of code, making them well-suited for large-scale software development projects. Examples of object-oriented languages include Java, C++, Python, and C#.

- **Key components of an E-R diagram include:**

1. *Classes:* Blueprints for creating objects, defining their attributes and behaviors.

2. *Objects:* Instances of classes, representing specific entities with unique data values.

3. *Encapsulation:* Bundling data and methods within a class, controlling access to data through methods.

4. *Inheritance:* Allowing new classes to inherit properties and behaviors from existing classes, promoting code reuse.

5. *Polymorphism:* Treating objects of different types through a common interface, enabling flexibility and extensibility.

6.*Abstraction:* Modeling real-world entities as simplified representations, focusing on essential features while hiding unnecessary details.

2.2 Why we use Object Oriented Than Procedure Oriented

Object-oriented programming (OOP) offers several advantages over procedure-oriented programming (POP):

1. *Modularity:* OOP promotes modular design by encapsulating data and behavior within objects. This makes code easier to understand, maintain, and reuse, as changes to one part of the codebase are less likely to impact other parts.

2. *Code Reusability:* Inheritance allows classes to inherit properties and behaviors from parent classes, enabling code reuse. This reduces redundancy and promotes efficient development by leveraging existing code.

3. *Flexibility and Extensibility:* Polymorphism enables objects of different types to be treated uniformly through a common interface. This promotes flexibility, allowing for easy extension and modification of code without affecting existing functionality.

4. *Encapsulation and Data Hiding:* OOP hides the implementation details of objects, exposing only necessary interfaces. This protects data integrity and prevents unauthorized access to data, enhancing security and reliability.

5. *Abstraction:* OOP allows developers to model real-world entities as abstract representations, focusing on essential attributes and behaviors while hiding unnecessary details. This simplifies complex systems and improves code readability.

6. *Scalability:* OOP provides a natural way to model complex systems, making it easier to scale up the size and complexity of software projects. Objects can be instantiated and reused as needed, facilitating the development of large-scale applications.

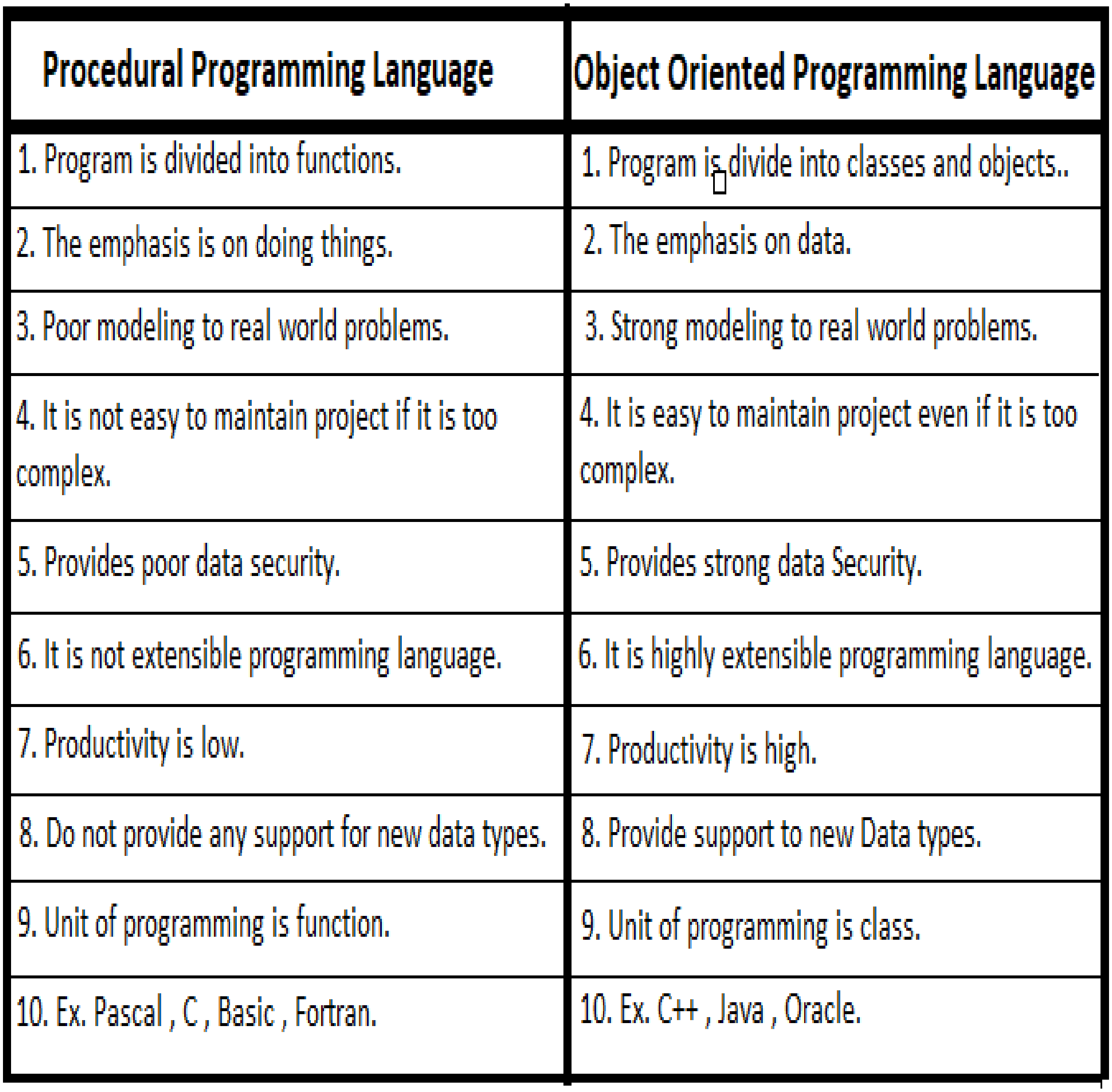
Overall, OOP promotes better code organization, reusability, maintainability, and scalability compared to POP, making it a preferred choice for modern software development.

2.3 Simple Java Program using Class and Object:-

```
// Define a class called "Rectangle"
class Rectangle {
// Define instance variables
double length;
double width;

// Constructor to initialize the object
public Rectangle(double length, double width) {
this.length = length;
this.width = width;
}

// Method to calculate area of the rectangle
public double calculateArea() {
return length * width;
}
}
// Main class
public class Main {
public static void main(String[] args) {
// Create an object of the Rectangle class
Rectangle myRectangle = new Rectangle(5.0, 3.0);
// Calculate the area of the rectangle
double area = myRectangle.calculateArea();
// Display the area of the rectangle
System.out.println("Area of the rectangle: " + area);
}
}
```

Output:- Area of the rectangle: 15.0

2.4 Difference Between object Oriented & Procedure oriented:-



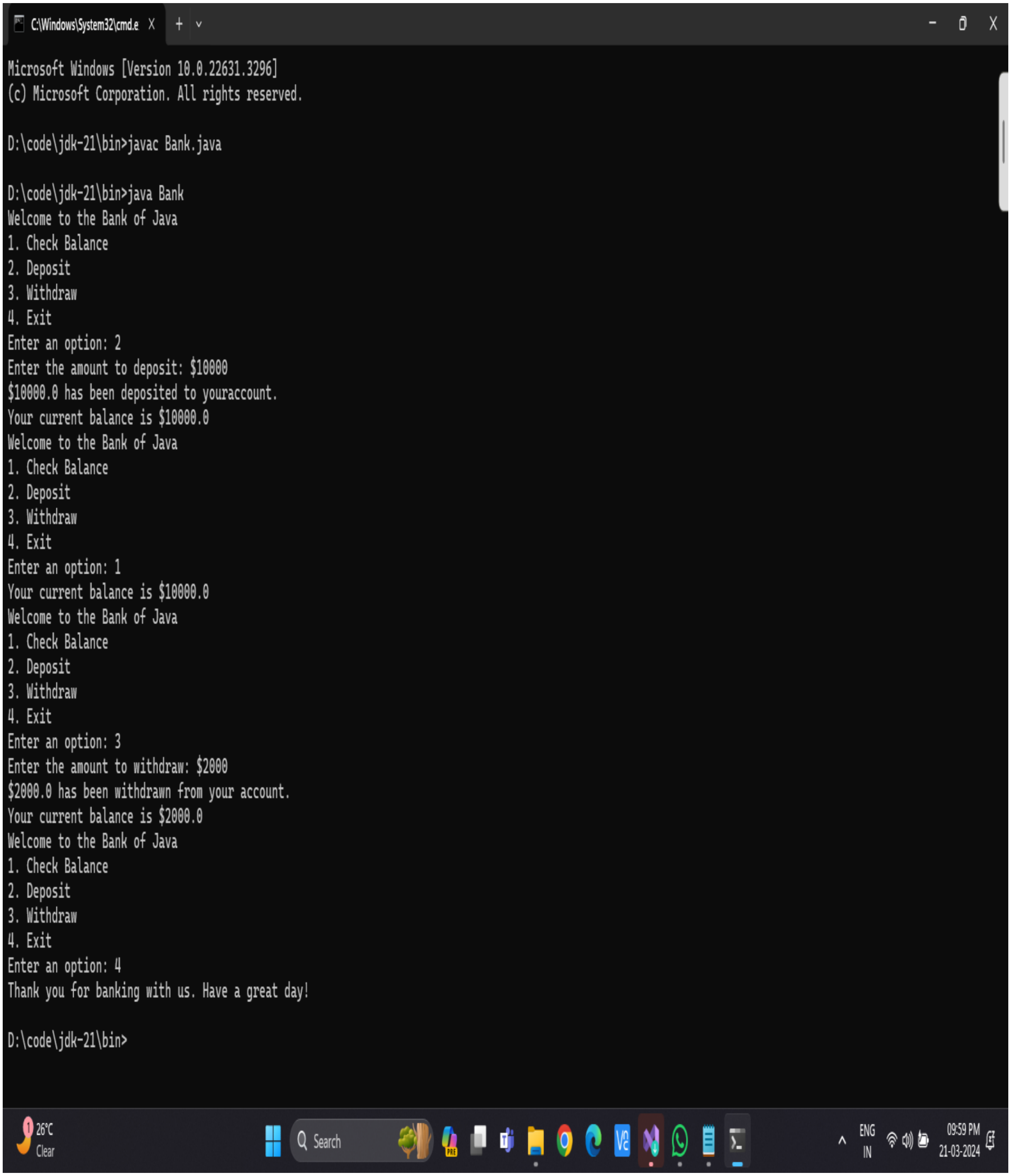
Procedural Programming Language	Object Oriented Programming Language
1. Program is divided into functions.	1. Program is divide into classes and objects..
2. The emphasis is on doing things.	2. The emphasis on data.
3. Poor modeling to real world problems.	3. Strong modeling to real world problems.
4. It is not easy to maintain project if it is too complex.	4. It is easy to maintain project even if it is too complex.
5. Provides poor data security.	5. Provides strong data Security.
6. It is not extensible programming language.	6. It is highly extensible programming language.
7. Productivity is low.	7. Productivity is high.
8. Do not provide any support for new data types.	8. Provide support to new Data types.
9. Unit of programming is function.	9. Unit of programming is class.
10. Ex. Pascal , C , Basic , Fortran.	10. Ex. C++ , Java , Oracle.

3. Bank Management System Program

```
import java.util.Scanner;

public class Bank {
  static double balance = 0;

  public static void main(String[] args)
 {
    Scanner scanner = new Scanner(System.in);
    int option = 0;
    while (option != 4)
{
      System.out.println("Welcome to the Bank of Java");
      System.out.println("1. Check Balance");
      System.out.println("2. Deposit");
      System.out.println("3. Withdraw");
      System.out.println("4. Exit");
      System.out.print("Enter an option: ");
      option = scanner.nextInt();

      switch (option)
     {
        case 1:
          checkBalance();
          break;
        case 2:
          deposit();
          break;
        case 3:
          withdraw();
    break;
   case 4:
    exit();
    break;
   default:
    System.out.println("Invalid option. Try again.");
    break;
  }
 }
}

public static void checkBalance()
{
 System.out.println("Your current balance is $" + balance);
}

public static void deposit()
{
 Scanner scanner = new Scanner(System.in);
 System.out.print("Enter the amount to deposit: $");
 double amount = scanner.nextDouble();
 balance += amount;
 System.out.println("$" + amount + " has been deposited to youraccount.");
 checkBalance();
}

public static void withdraw()
{
 Scanner scanner = new Scanner(System.in);
 System.out.print("Enter the amount to withdraw: $");
 double amount = scanner.nextDouble();
```

```
    if (amount > balance)
   {
      System.out.println("Insufficient funds.");
    }
    else
     {
      balance = amount;
      System.out.println("$" + amount + " has been withdrawn from your account.");
    }
    checkBalance();
  }

  public static void exit()
   {
    System.out.println("Thank you for banking with us. Have a great day!");
    }
}
```

4.Output of The Program



```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

D:\code\jdk-21\bin>javac Bank.java

D:\code\jdk-21\bin>java Bank
Welcome to the Bank of Java
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter an option: 2
Enter the amount to deposit: $10000
$10000.0 has been deposited to youraccount.
Your current balance is $10000.0
Welcome to the Bank of Java
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter an option: 1
Your current balance is $10000.0
Welcome to the Bank of Java
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter an option: 3
Enter the amount to withdraw: $2000
$2000.0 has been withdrawn from your account.
Your current balance is $2000.0
Welcome to the Bank of Java
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter an option: 4
Thank you for banking with us. Have a great day!

D:\code\jdk-21\bin>
```

5. Conclusion

In summary, the Java-based bank management system microproject serves as a versatile and practical tool for facilitating efficient banking operations. By employing Java's robust features and database management capabilities, it offers a user-friendly interface for managing accounts, transactions, and user authentication. The project underscores the significance of object-oriented programming principles in developing scalable and secure software solutions for the financial sector. Moving forward, continual refinement and adaptation of the system can further enhance its functionality and meet the evolving needs of modern banking institutions.