

Academic Year: 2023-24

Semester: II

Class: FYMCA

Course Code: MC506

Course Name: Java Programming

LABORATORY ASSESSMENT

CLASS: FY

SEM: II

BRANCH: MCA

ACADEMIC YEAR: 2023-2024

COURSE: Java Programming

COURSE CODE: MC506

STUDENT NAME: Vivek Tiwari

ROLL NO: 2023510059

Experiment No. 1.3

Aim: Fundamentals of JAVA programming -II using Eclipse IDE

CO Mapping – CO 1

Objective:

- To understand declaration of Classes, and Methods with its all features such as Constructors, Access Specifier
- To understand Classes, Instance variables, Methods, Constructors, Access
- Specifiers as basic fundamentals
- Implement Abstract Classes and Wrapper Classes for given problem statement
- Design and implement Inheritance, Polymorphism in JAVA
- Demonstrate Use of Static, final, super and this keyword
- Demonstrate creating user defined package, Access control protection,
- Defining interface, implementing interface

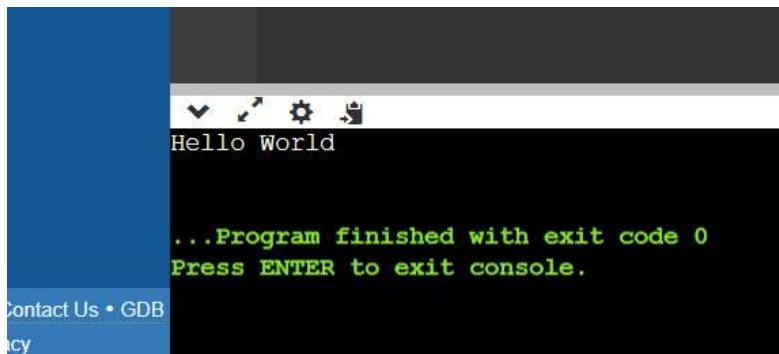
Lab Exercise:

1. W.A.P on to print "Hello World ".

Code:

```
package fymca_59;
public class one {
    public static void main(String[] args) {
        System.out.print("Hello World!");
    }
}
```

Output:



2. Create a class cart in a shopping store. The class should have Product Name, Product Id, Product purchase date, Product Category (Apply respective data structure). Implement a method to add Total Products purchased and Final values of each day purchase. Use appropriate Objects and Constructors wherever needed.

Code:

```
package fymca_59;
import java.util.*;
public class Cart {
    private String productName;
    private int productId;
    private Date purchaseDate;
    private String productCategory;
    public Cart(String productName, int productId, Date purchaseDate, String
productCategory) {
        this.productName = productName;
        this.productId = productId;
        this.purchaseDate = purchaseDate;
        this.productCategory = productCategory;
    }
    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }
}
```

```

    }

    public int getProductId() {
        return productId;
    }

    public void setProductId(int productId) {
        this.productId = productId;
    }

    public Date getPurchaseDate() {
        return purchaseDate;
    }

    public void setPurchaseDate(Date purchaseDate) {
        this.purchaseDate = purchaseDate;
    }

    public String getProductCategory() {
        return productCategory;
    }

    public void setProductCategory(String productCategory) {
        this.productCategory = productCategory;
    }

    public static int addTotalProductsPurchased(List<Cart> cartList) {
        return cartList.size();
    }

    public static Map<Date, Integer>
    getFinalValuesOfEachDayPurchase(List<Cart> cartList) {
        Map<Date, Integer> finalValues = new HashMap<>();
        for (Cart cart : cartList) {
            Date date = cart.getPurchaseDate();

            int count = finalValues.getOrDefault(date, 0);
            finalValues.put(date, count + 1);
        }
        return finalValues;
    }

    public static void main(String[] args) {
        Cart cart1 = new Cart("Laptop", 1059, new Date(), "Electronics");
        Cart cart2 = new Cart("Mobile Phone", 1002, new Date(),
            "Electronics");
        Cart cart3 = new Cart("Dress", 2059, new Date(), "Clothing");
        Cart cart4 = new Cart("Shoes", 2002, new Date(), "Footwear");
        Cart cart5 = new Cart("Earphone", 2000, new Date(), "Electronics");
        List<Cart> cartList = new ArrayList<>();
        cartList.add(cart1);
        cartList.add(cart2);
        cartList.add(cart3);
        cartList.add(cart4);
        cartList.add(cart5);
        System.out.println("Purchased Products Information:");
        for (Cart cart : cartList) {
            System.out.println("Product Name: " + cart.getProductName());
            System.out.println("Product ID: " + cart.getProductId());
            System.out.println("Purchase Date: " + cart.getPurchaseDate());
        }
    }
}

```

```

System.out.println("Product Category: " +
cart.getProductCategory());
    System.out.println(" -----"); }
    int totalProductsPurchased = addTotalProductsPurchased(cartList);
System.out.println("Total products purchased: " +
totalProductsPurchased);
    Map<Date, Integer> finalValues =
getFinalValuesOfEachDayPurchase(cartList);
System.out.println("Final values of each day purchase: " +
finalValues);
}
}

```

Output:

```

Purchased Products Information:
Product Name: Laptop
Product ID: 1001
Purchase Date: Wed Feb 21 06:31:22 GMT 2024
Product Category: Electronics
-----
Product Name: Mobile Phone
Product ID: 1002
Purchase Date: Wed Feb 21 06:31:22 GMT 2024
Product Category: Electronics
-----
Product Name: Dress
Product ID: 2001
Purchase Date: Wed Feb 21 06:31:22 GMT 2024
Product Category: Clothing
-----
Product Name: Shoes
Product ID: 2002
Purchase Date: Wed Feb 21 06:31:22 GMT 2024
Product Category: Footwear
-----
Product Name: Earphone
Product ID: 2000
Purchase Date: Wed Feb 21 06:31:22 GMT 2024
Product Category: Electronics
-----
Total products purchased: 5
Final values of each day purchase: {Wed Feb 21 06:31:22 GMT 2024=5}

```

3. Create a class Laptop with specifications and assign suitable data types to its feature

Make use of suitable methods and Constructors and display the output. Code:

```

package fymca_59;
import java.util.Scanner;
public class Laptop {
    private String brand;
    private String model;
    private double screenSize;
    private int ramSizeGB;
    private double storageSizeGB;
    private String processor;
    private boolean touchscreenEnabled;
    public Laptop(String brand, String model, double screenSize, int ramSizeGB,
double storageSizeGB, String processor, boolean touchscreenEnabled) {
this.brand = brand;
    this.model = model;

```

```

    this.screenSize = screenSize;
    this.ramSizeGB = ramSizeGB;
    this.storageSizeGB = storageSizeGB;
    this.processor = processor;
    this.touchscreenEnabled = touchscreenEnabled;
}
public void displaySpecifications() {
System.out.println("Laptop Specifications:");
System.out.println("Brand: " + brand);
System.out.println("Model: " + model);
System.out.println("Screen Size: " + screenSize + " inches");
System.out.println("RAM Size: " + ramSizeGB + " GB");
System.out.println("Storage Size: " + storageSizeGB + " GB");
System.out.println("Processor: " + processor);
    System.out.println("Touchscreen Enabled: " + (touchscreenEnabled ? "Yes" :
"No"));
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.println("Enter Laptop Specifications:");
System.out.print("Brand: ");
String brand = scanner.nextLine();
System.out.print("Model: ");
String model = scanner.nextLine();
System.out.print("Screen Size (in inches): ");
double screenSize = scanner.nextDouble();
System.out.print("RAM Size (in GB): ");
int ramSizeGB = scanner.nextInt();
System.out.print("Storage Size (in GB): ");
double storageSizeGB = scanner.nextDouble();
scanner.nextLine();
System.out.print("Processor: ");
String processor = scanner.nextLine();
System.out.print("Is Touchscreen Enabled (true/false): ");
boolean touchscreenEnabled = scanner.nextBoolean();

Laptop laptop = new Laptop(brand, model, screenSize, ramSizeGB,
storageSizeGB, processor, touchscreenEnabled);
laptop.displaySpecifications();
    scanner.close();
}
}

```

Output:

```

Enter Laptop Specifications:
Brand: Acer
Model: A315
Screen Size (in inches): 14.6
RAM Size (in GB): 512
Storage Size (in GB): 1000
Processor: AMD Ryzen
Is Touchscreen Enabled (true/false): false
Laptop Specifications:
Brand: Acer
Model: A315
Screen Size: 14.6 inches
RAM Size: 512 GB
Storage Size: 1000.0 GB
Processor: AMD Ryzen
Touchscreen Enabled: No

```

4. Define a class subject. Take input string as name "Java" Show the o/p as J.V. using conversion to char []. Then using CharAt() show the o/p.

Code:

```

package fymca_59;
import java.util.Scanner;
public class Subject {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the subject name: ");
        String subjectName = scanner.nextLine();
        scanner.close();
        char[] charArray = subjectName.toCharArray();
        System.out.println("Output using conversion to char[]:");
        System.out.println(charArray[0] + "." + charArray[charArray.length - 2] + ".");
        System.out.println("Output using CharAt():");
        System.out.println(subjectName.charAt(0) + "." +
            subjectName.charAt(subjectName.length() - 2) + ".");
    }
}

```

Output:

```

Enter the subject name: JAVA
Output using conversion to char[]:
J.V.
Output using CharAt():
J.V.

```

5. W.A.P based on Is- A relationship in a context of Organization which have Employee & Programmer "Is-A" relationship.

Code:

```

package fymca_59;
class Employee {
    private String name;

```

```

private int employeeId;
private double salary;
public Employee(String name, int employeeId, double salary) {
    this.name = name;
    this.employeeId = employeeId;
    this.salary = salary;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public int getEmployeeId() {
    return employeeId;
}
public void setEmployeeId(int employeeId) {
    this.employeeId = employeeId;
}
public double getSalary() {
    return salary;
}
public void setSalary(double salary) {
    this.salary = salary;
}
public void displayEmployeeInfo() {
    System.out.println("Employee Name: " + name);
    System.out.println("Employee ID: " + employeeId);
    System.out.println("Salary: " + salary);
}
}
class Programmer extends Employee {
private String programmingLanguage;
public Programmer(String name, int employeeId, double salary,
String programmingLanguage) {
    super(name, employeeId, salary);
    this.programmingLanguage = programmingLanguage;
}
public String getProgrammingLanguage() {
    return programmingLanguage;
}
public void setProgrammingLanguage(String programmingLanguage)
{ this.programmingLanguage = programmingLanguage;
}
public void displayProgrammerInfo() {
    System.out.println("Programmer Name: " + getName());
    System.out.println("Employee ID: " + getEmployeeId());
    System.out.println("Salary: " + getSalary());

    System.out.println("Programming Language: " + programmingLanguage);
}
}
public class Main {
public static void main(String[] args) {
    Employee employee = new Employee("Vivek", 1001, 50000);

```

```
System.out.println("Employee Information:");
employee.displayEmployeeInfo();
System.out.println();
Programmer programmer = new Programmer("Vaibhav", 2001, 70000, "Java");
System.out.println("Programmer Information:");
programmer.displayProgrammerInfo();
}
}
```

Output:

```
Employee Information:
Employee Name: Vivek
Employee ID: 1001
Salary: 50000.0

Programmer Information:
Programmer Name: Vaibhav
Employee ID: 2001
Salary: 70000.0
Programming Language: Java
```

6. W.A.P and explain the "Has-A" relationship.

Code:

```
package fymca_59;
class Address {
    private String street;
    private String city;
    private String state;
    private String zipcode;
    public Address(String street, String city, String state, String zipcode)
    { this.street = street;
      this.city = city;
      this.state = state;
      this.zipcode = zipcode;
    }
    public String getStreet() {
        return street;
    }
    public void setStreet(String street) {
```



```

    this.street = street;
}
public String getCity() {
    return city;
}
public void setCity(String city) {
    this.city = city;
}

public String getState() {
    return state;
}
public void setState(String state) {
    this.state = state;
}
public String getZipcode() {
    return zipcode;
}
public void setZipcode(String zipcode) {
    this.zipcode = zipcode;
}
public void displayAddress() {
    System.out.println("Address: " + street + ", " + city + ", " + state + ", " +
        zipcode);
}
}
class Employee {
    private String name;
    private int employeeId;
    private double salary;
    private Address address;
    public Employee(String name, int employeeId, double salary, Address address)
    { this.name = name;
      this.employeeId = employeeId;
      this.salary = salary;
      this.address = address;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getEmployeeId() {
        return employeeId;
    }
    public void setEmployeeId(int employeeId) {
        this.employeeId = employeeId;
    }
    public double getSalary() {
        return salary;
    }
    public void setSalary(double salary) {
        this.salary = salary;
    }
    public Address getAddress() {

```

```

    return address;
}
public void setAddress(Address address) {
    this.address = address;
}
public void displayEmployeeInfo() {
    System.out.println("Employee Name: " + name);
    System.out.println("Employee ID: " + employeeId);

    System.out.println("Salary: " + salary);
    address.displayAddress();
}
}
public class Main1 {
    public static void main(String[] args) {
        Address address = new Address("Shree Pooja apartment", "Virar", "West",
"459303");
        Employee employee = new Employee("Vivek", 1001, 500000, address);
        System.out.println("Employee Information:");
        employee.displayEmployeeInfo();
    }
}

```

Output:

```

Employee Information:
Employee Name: Vivek
Employee ID: 1001
Salary: 500000.0
Address: Shree Pooja apartment, Virar, West, 459303

```

7. W.A.P based on the concept of Inheritance using Shape class of various geometrical figures. Calculate Area for different shapes

Code:

```

package fymca_59;
import java.util.Scanner;
import java.lang.Math;
class Shape {

    public double calculateArea() {
        return 0.0;
    }
}
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}

```

```

    }
    class Rectangle extends Shape {
        private double length;
        private double width;
        public Rectangle(double length, double width) {
            this.length = length;
            this.width = width;
        }

        @Override
        public double calculateArea() {
            return length * width;
        }
    }

    class Triangle extends Shape {
        private double base;
        private double height;
        public Triangle(double base, double height) {
            this.base = base;
            this.height = height;
        }

        @Override
        public double calculateArea() {
            return 0.5 * base * height;
        }
    }

    public class Main1 {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the radius of the circle: ");
            double circleRadius = scanner.nextDouble();
            Circle circle = new Circle(circleRadius);
            System.out.print("Enter the length of the rectangle: ");
            double rectangleLength = scanner.nextDouble();
            System.out.print("Enter the width of the rectangle: ");
            double rectangleWidth = scanner.nextDouble();
            Rectangle rectangle = new Rectangle(rectangleLength,
            rectangleWidth);
            System.out.print("Enter the base of the triangle: ");
            double triangleBase = scanner.nextDouble();
            System.out.print("Enter the height of the triangle: ");
            double triangleHeight = scanner.nextDouble();
            Triangle triangle = new Triangle(triangleBase, triangleHeight);
            scanner.close();
            double circleArea = circle.calculateArea();
            double rectangleArea = rectangle.calculateArea();
            double triangleArea = triangle.calculateArea();
            System.out.println("Area of Circle: " + circleArea);
            System.out.println("Area of Rectangle: " + rectangleArea);
            System.out.println("Area of Triangle: " + triangleArea); }
    }

```

Output:

```
Enter the radius of the circle: 7
Enter the length of the rectangle: 10
Enter the width of the rectangle: 5
Enter the base of the triangle: 20
Enter the height of the triangle: 6
Area of Circle: 153.93804002589985
Area of Rectangle: 50.0
Area of Triangle: 60.0
```

8. W.A.P to define a Class "Animal" and a subclass "Lion"

Code:

```
class Animal {
    private String name;
    private int age;
    private String sound;
    public Animal(String name, int age, String sound) {
        this.name = name;
        this.age = age;
        this.sound = sound;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public void makeSound() {
        System.out.println(sound);
    }
}
class Lion extends Animal {
    private String type;
    public Lion(String name, int age, String sound, String type) {
        super(name, age, sound);
        this.type = type;
    }
    public String getType() {
        return type;
    }
}
public class Main {
    public static void main(String[] args) {
```

```

Lion lion = new Lion("BroLion", 5, "Roar", "African");
System.out.println("Name: " + lion.getName());
System.out.println("Age: " + lion.getAge());
System.out.println("Type: " + lion.getType());
System.out.print("Sound: ");
lion.makeSound();
}
}

```

Output:

```

Name: BroLion
Age: 5
Type: African
Sound: Roar

```

9. Find out the Error in the following program- Inheritance using Bank Class ,rectify if any.

```

import java.lang.*;
import java.util.*; class Account
{
String Name; int acno; double balance; double checkbal()
{
return balance;
}
}
class AccountEx extends Account
{
AccountEx (String Name,int a,double b)
{
this.Name=Name; acno=a; balance=b;
}
void withdraw(double amt)
{
balance=balance-amt; if(balance < 500)
{
System.out.println("Can't Withdraw minimum balance should be greater than 500");
}
}
void deposit(double amt)
{

```

```

balance=balance+amt;
}
void transfer(Account b, double k)
{
balance=balance-k; if(balance<500.00)
{
transfer");
"+balance); b.balance);
}
}
else
System.out.println("You don't have sufficient balance to
b.balance=b.balance+k; System.out.println("acno:"+acno+" "+"balance is"+"
System.out.println("acno:"+b.acno+" "+"balance is"+"
}
class AccountInheritance

{
public static void main(String args[])
{
AccountEx s = new AccountEx("Pradnya",1,55000.00); System.out.println("The balance
is:"+s.checkbal());
System.out.println("Enter the withdrawing amt"); Scanner sc = new Scanner(System.in);
double i =
sc.nextDouble(); s.withdraw(i);
System.out.println("The balance is:"+s.checkbal());
System.out.println("Enter the Deposit amt"); Scanner sc1 = new Scanner(System.in); double
j =
sc1.nextDouble();
s.deposit(j);
System.out.println("The balance is:"+s.checkbal()); AccountEx b = new
AccountEx("Nameeta",2,65000.00);
System.out.println("Enter amt to transfer:"); Scanner sc2 = new Scanner(System.in); double
k =
sc2.nextDouble();
s.transfer(b,k);
}
}

```

Code:

```

package mca_59;
import java.util.*;

```

```

class Account {
    String Name;
    int acno;
    double balance;

    double checkbal() {
        return balance;
    }
}

class AccountEx extends Account {
    AccountEx(String Name, int a, double b) {
        this.Name = Name;
        acno = a;
        balance = b;
    }

    void withdraw(double amt) {
        balance = balance - amt;
        if (balance < 500) {
            System.out.println("Can't Withdraw minimum balance should be greater than 500");
        }
    }

    void deposit(double amt) {
        balance = balance + amt;
    }

    void transfer(AccountEx b, double k) {
        balance = balance - k;
        if (balance < 500.00) {
            System.out.println("Insufficient balance for transfer");
            System.out.println("Your balance: " + balance);
            System.out.println("Receiver's balance: " + b.balance); } else
        {
            b.balance = b.balance + k;
            System.out.println("Transfer successful");
            System.out.println("Your balance after transfer: " + balance);
            System.out.println("Receiver's balance after transfer: " + b.balance); }
    }
}

public class Main1 {
    public static void main(String args[]) {
        AccountEx s = new AccountEx("Pradnya", 1, 55000.00);
        System.out.println("The balance is: " + s.checkbal());
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the withdrawing amt");
        double i = sc.nextDouble();
        s.withdraw(i);
        System.out.println("The balance is: " + s.checkbal());

        System.out.println("Enter the Deposit amt");
    }
}

```

```

double j = sc.nextDouble();
s.deposit(j);
System.out.println("The balance is: " + s.checkbal());

AccountEx b = new AccountEx("Nameeta", 2, 65000.00);
System.out.println("Enter amt to transfer:");
double k = sc.nextDouble();
s.transfer(b, k);
sc.close();
}
}

```

Output:

```

The balance is: 55000.0
Enter the withdrawing amt
1000
The balance is: 54000.0
Enter the Deposit amt
10000
The balance is: 64000.0
Enter amt to transfer:
500
Transfer successful
Your balance after transfer: 63500.0
Receiver's balance after transfer: 65500.0

```

10. W.A.P to print command line arguments using for loop

Code:

```

package mac_59;
public class Main1 {
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++) {
            System.out.println("Argument " + (i + 1) + ": " + args[i]);
        }
    }
}

```

Output:

```

Argument 1: java
Argument 2: PrintCommandLineArguments
Argument 3: arg1
Argument 4: arg2
Argument 5: arg3

```

11. W.A.P to create a Class "Currency Converter" to convert Rupees into Different Currencies

Code:

```

package fymca_59;
import java.util.Scanner;

```



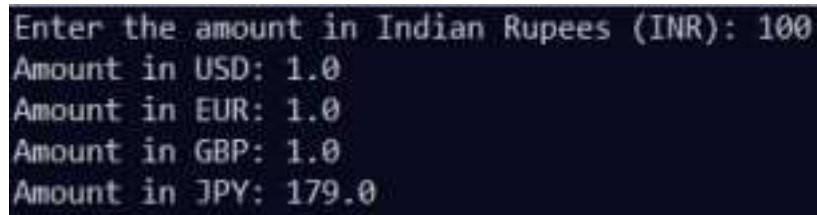
```

public class CurrencyConverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the amount in Indian Rupees (INR): ");
        double amountInINR = scanner.nextDouble();
        double usdRate = 0.59;
        double eurRate = 0.59;

        double gbpRate = 0.59;
        double jpyRate = 1.79;
        double amountInUSD = amountInINR * usdRate;
        double amountInEUR = amountInINR * eurRate;
        double amountInGBP = amountInINR * gbpRate;
        double amountInJPY = amountInINR * jpyRate;
        System.out.println("Amount in USD: " + amountInUSD);
        System.out.println("Amount in EUR: " + amountInEUR);
        System.out.println("Amount in GBP: " + amountInGBP);
        System.out.println("Amount in JPY: " + amountInJPY);
        scanner.close();
    }
}

```

Output:



```

Enter the amount in Indian Rupees (INR): 100
Amount in USD: 1.0
Amount in EUR: 1.0
Amount in GBP: 1.0
Amount in JPY: 179.0

```

12. W.A.P to generate Lottery Numbers (1 to 49)

Code:

```

package fymca_59;
import java.util.Arrays;
import java.util.Random;
public class LotteryNumberGenerator {
    public static void main(String[] args) {
        int[] lotteryNumbers = generateLotteryNumbers(6);
        System.out.println("Generated Lottery Numbers: " +
            Arrays.toString(lotteryNumbers));
    }
    public static int[] generateLotteryNumbers(int count) {
        int[] numbers = new int[count];
        Random random = new Random();

        for (int i = 0; i < count; i++) {
            int randomNumber;
            do {
                randomNumber = random.nextInt(49) + 1;
            } while (contains(numbers, randomNumber));
            numbers[i] = randomNumber;
        }

        return numbers;
    }
}

```

```

    }
    public static boolean contains(int[] arr, int number) {
    for (int num : arr) {
    if (num == number) {
    return true;
    }

    }
    return false;
    }
}

```

Output:



Generated Lottery Numbers: [20, 40, 33, 28, 12, 6]

13. W.A.P to generate a random sequence of capital letters such that in one line only 6 letters can be seen.

Code:

```

package fymca_59;

import java.util.Random;

public class Main {
    public static void main(String[] args) {
        generateAndPrintRandomSequence(30);
    }
    public static void generateAndPrintRandomSequence(int length) {
        Random random = new Random();
        int lettersPerLine = 6;
        int count = 0;
        for (int i = 0; i < length; i++) {
            char letter = (char) (random.nextInt(26) + 'A');
            System.out.print(letter + " ");
            count++;
            if (count == lettersPerLine) {
                System.out.println();
                count = 0;
            }
        }
    }
}

```

Output:



J	J	J	V	F	G
H	B	F	N	F	V
I	W	M	G	G	N
P	S	F	K	G	F
Y	A	V	I	I	H

14. W.A.P using Constructor and Destructor for a Class "Box " using depth, height and

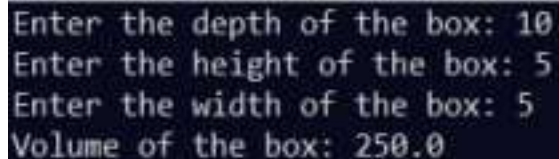
width as parameters for Volume.

Code:

```
package fymca_59;
import java.util.Scanner;
public class Box {

    private double depth;
    private double height;
    private double width;
    public Box(double depth, double height, double width) {
        this.depth = depth;
        this.height = height;
        this.width = width;
    }
    public double calculateVolume() {
        return depth * height * width;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the depth of the box: ");
        double depth = scanner.nextDouble();
        System.out.print("Enter the height of the box: ");
        double height = scanner.nextDouble();
        System.out.print("Enter the width of the box: ");
        double width = scanner.nextDouble();
        Box myBox = new Box(depth, height, width);
        System.out.println("Volume of the box: " + myBox.calculateVolume());
        scanner.close();
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The text is as follows:

```
Enter the depth of the box: 10
Enter the height of the box: 5
Enter the width of the box: 5
Volume of the box: 250.0
```

15. Identify the error in the give program class Circle

```
{
    private double radius;
    String color;
    /* Constructor defination*/
    public Circle(double r , String c)
    {
        radius = r; color = c;
    }
    public double getRadius()
    {
```

```

return radius;
}
public double findArea()
{

return radius*radius*Math.PI;
}
} // close circle class defination
public class TestObject
{
public static void main(String args[])
{
Circle myc = new Circle(5.0,"blue"); printCircle(myc);
colorCircle(myc,"black");
printCircle(myc);
}
public static void colorCircle(Circle c , String color)
1
{
c.color = color;
}
public static void printCircle(Circle c)
{
System.out.println("The Area of the Circle of the
radius"+c.getRadius()+" is
"+c.findArea()); System.out.println("The Color of the circle is:
"+c.color);
}
}

```

Code:

```

package fymca_59;
class Circle {
    private double radius;
    String color;
    public Circle(double r , String c) {
        radius = r;
        color = c;
    }
    public double getRadius() {
        return radius;
    }
    public double findArea() {
        return radius * radius * Math.PI;
    }
}
public class Main {
    public static void main(String args[]) {

```

```

Circle myc = new Circle(5.0, "blue");
printCircle(myc);
colorCircle(myc, "black");
printCircle(myc);
}

```

```

public static void colorCircle(Circle c, String color) {
    c.color = color;
}
public static void printCircle(Circle c) {
    System.out.println("The Area of the Circle of the radius " + c.getRadius() + " is " + c.findArea());
    System.out.println("The Color of the circle is: " + c.color); }
}

```

Output:

```

The Area of the Circle of the radius 5.0 is 78.53981633974483
The Color of the circle is: blue
The Area of the Circle of the radius 5.0 is 78.53981633974483
The Color of the circle is: black

```

16. Execute the code and Demonstrate the significance of how Super Keyword is used in the Program. class Parent

```

{
int a;
public Parent(int val)
{
a = val;
}
}
public class Child extends Parent
{
int b=9,c; public Child()
{
super(5);
}
public void add()
{
c = a+b;
}
public void show()
{
System.out.println("&quot;Addition is: &quot;+c);
}
public static void main(String args[])

```

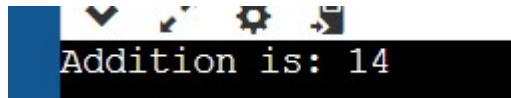
```
{  
Child ch = new Child(); ch.add();  
ch.show();
```

```
1  
}  
}
```

Code:

```
package fymca_59;  
class Parent {  
int a;  
    public Parent(int val) {  
        a = val;  
    }  
}  
public class Child extends Parent {  
    int b = 9;  
    int c;  
    public Child() {  
        super(5);  
    }  
    public void add() {  
        c = a + b;  
    }  
    public void show() {  
        System.out.println("Addition is: " + c);  
    }  
    public static void main(String args[]) {  
        Child ch = new Child();  
        ch.add();  
        ch.show();  
    }  
}
```

Output:

A screenshot of a terminal window with a dark background. At the top, there are four small icons: a blue square, a white checkmark, a white gear, and a white document. Below these icons, the text "Addition is: 14" is displayed in a yellow, monospaced font.

In the given program, the super keyword is used in the Child class constructor to invoke the constructor of the superclass Parent.

17. W.A.P to create Bank Account Class using Interfaces.

Code:

```
package fymca_59;  
import java.util.Scanner;  
  
//Interface defining common operations for bank accounts  
interface BankAccountOperations {  
    void deposit(double amount);  
    void withdraw(double amount);  
}
```

```

    double getBalance();
}

//Concrete implementation of the BankAccount interface
class BankAccount implements BankAccountOperations {
    private double balance;

    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }
    @Override
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }
    @Override
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance");
        }
    }
    @Override
    public double getBalance() {
        return balance;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();
        BankAccountOperations account = new BankAccount(initialBalance);
        System.out.print("Enter deposit amount: ");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);
        System.out.print("Enter withdrawal amount: ");
        double withdrawalAmount = scanner.nextDouble();
        account.withdraw(withdrawalAmount);
        System.out.println("Final Balance: " + account.getBalance());
        scanner.close();
    }
}

```

Output:

```
Enter initial balance: 1000000
Enter deposit amount: 500000
Deposited: 500000.0
Enter withdrawal amount: 153400
Withdrawn: 153400.0
B Final Balance: 1346600.0
```

18. W.A.P to create Bank Account Class using Abstract Classes.

Code:

```
package fymca_59;
import java.util.Scanner;

abstract class BankAccount {
    protected double balance;
    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }
    public abstract void deposit(double amount);
    public abstract void withdraw(double amount);
    public double getBalance() {
        return balance;
    }
}
class SavingsAccount extends BankAccount {
    public SavingsAccount(double initialBalance) {
        super(initialBalance);
    }
    @Override
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }
    @Override
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance");
        }
    }
}
class CurrentAccount extends BankAccount {
    public CurrentAccount(double initialBalance) {
        super(initialBalance);
    }
    @Override
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }
    @Override
```



```

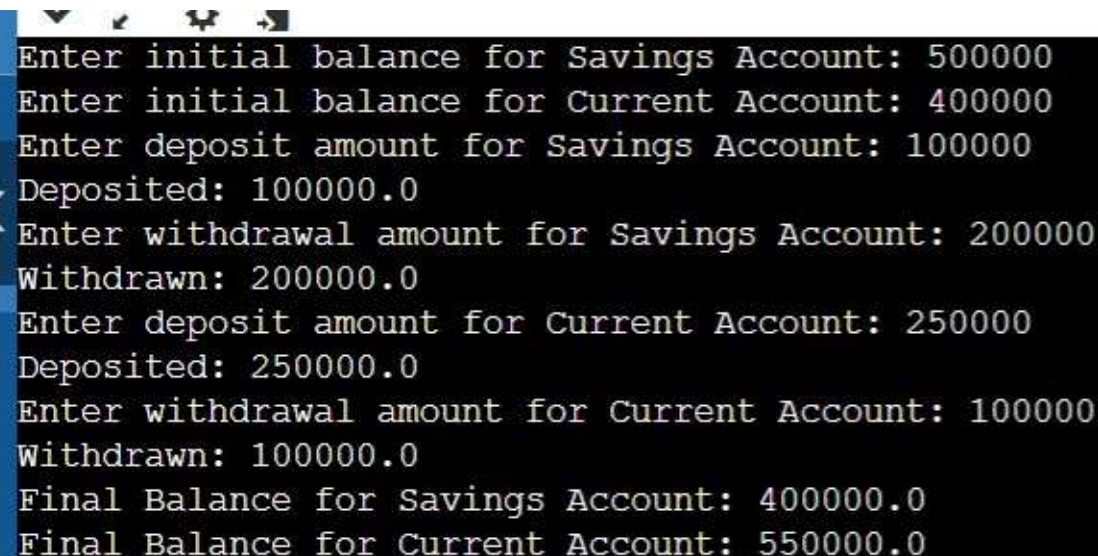
    public void withdraw(double amount) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter initial balance for Savings Account: ");
        double initialBalanceSavings = scanner.nextDouble();
        BankAccount savingsAccount = new SavingsAccount(initialBalanceSavings);
        System.out.print("Enter initial balance for Current Account: "); double
        initialBalanceCurrent = scanner.nextDouble();
        BankAccount currentAccount = new CurrentAccount(initialBalanceCurrent);
        System.out.print("Enter deposit amount for Savings Account: ");

        double depositAmountSavings = scanner.nextDouble();
        savingsAccount.deposit(depositAmountSavings);
        System.out.print("Enter withdrawal amount for Savings Account: ");
        double withdrawalAmountSavings = scanner.nextDouble();
        savingsAccount.withdraw(withdrawalAmountSavings);
        System.out.print("Enter deposit amount for Current Account: ");
        double depositAmountCurrent = scanner.nextDouble();
        currentAccount.deposit(depositAmountCurrent);
        System.out.print("Enter withdrawal amount for Current Account: ");
        double withdrawalAmountCurrent = scanner.nextDouble();
        currentAccount.withdraw(withdrawalAmountCurrent);
        System.out.println("Final Balance for Savings Account: " +
        savingsAccount.getBalance());
        System.out.println("Final Balance for Current Account: " +
        currentAccount.getBalance());
        scanner.close();
    }
}

```

Output:



```

Enter initial balance for Savings Account: 500000
Enter initial balance for Current Account: 400000
Enter deposit amount for Savings Account: 100000
Deposited: 100000.0
Enter withdrawal amount for Savings Account: 200000
Withdrawn: 200000.0
Enter deposit amount for Current Account: 250000
Deposited: 250000.0
Enter withdrawal amount for Current Account: 100000
Withdrawn: 100000.0
Final Balance for Savings Account: 400000.0
Final Balance for Current Account: 550000.0

```

19. Execute the following code and demonstrate how the concept of Wrapper class can be implemented in the program.

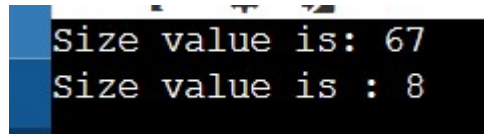
```
class Abs
{
private int size=8; /* protect the instance variables */
/* provide public getters and setters method */ public int getSize()
{
return size;
}
public void setSize(int newSize)
{
size = newSize;
}
} // close Abs class defination public class Abstract

{
public static void main(String args[])
{
Abs a = new Abs();
// a.size = 8; /* can not accessed instance variable ( Encapsulted) */
a.setSize(67);
System.out.println(&quot;Size value is: &quot;+a.getSize()); Abs a1 = new Abs();
System.out.println(&quot;Size value is : &quot;+a1.getSize());
}
}
```

Code:

```
package fymca_59;
class Abs {
    private int size = 8;
    public int getSize() {
        return size;
    }
    public void setSize(int newSize) {
        size = newSize;
    }
}
public class Main {
    public static void main(String args[]) {
        Abs a = new Abs();
        Integer newSize = 67;
        a.setSize(newSize);
        Integer size = a.getSize();
        System.out.println("Size value is: " + size);
        Abs a1 = new Abs();
        System.out.println("Size value is : " + a1.getSize());
    }
}
```

Output:



```
Size value is: 67
Size value is : 8
```

The provided code demonstrates encapsulation, where the instance variable `size` is encapsulated within the `Abs` class and accessed through public getter and setter methods. However, we can also illustrate the concept of wrapper classes by using them to modify the `size` attribute of the `Abs` class.

Wrapper classes in Java are used to convert primitive data types into objects (boxing) and vice versa (unboxing). In this case, we'll use the `Integer` wrapper class to set and retrieve the `size` attribute of the `Abs` class.

20. Demonstrate the significance of Package in the Program package MyPackage1;

```
//import java.io.*; import java.util.*;
```

```
//Factorial
```

```
public class Factorial
```

```
{
```

```
public void Factorial()
```

```
{
```

```
System.out.println("&quot;Enter Number To Find Factorial:&quot;); Scanner sc = new  
Scanner(System.in);
```

```
int n=sc.nextInt(); int f=1;
```

```
for(int i=n;i>0;i--)
```

```
{
```

```
f=f*i;
```

```
}
```

```
System.out.println("&quot;\nFactorial of &quot;+ n +&quot; is:&quot;+f);
```

```
}
```

```
}
```

```
//Maxinum
```

```
public class MaxNum
```

```
{
```

```
public void MaxNum()
```

```
{
```

```
System.out.println("&quot;\n1st number : &quot;); Scanner sc = new Scanner(System.in);
```

```
int mn1=sc.nextInt();
```

```
System.out.println("&quot;2nd number : &quot;); int mn2=sc.nextInt();
```

```
1
```

```
if(mn1>=mn2)
```

```

{
}
else
{
}
System.out.println(&quot;\nMax Num is 1st Number&quot;);
System.out.println(&quot;\nMax Num is 2nd Number&quot;);
}
}
//Rectangle
package MyPackage3;
import java.util.*; public class Rectangle
{
public void RectArea()
{

System.out.println(&quot;\nEnter Length of the Rectangle :&quot;); Scanner sc = new
Scanner(System.in);
int l=sc.nextInt();
System.out.println(&quot;Enter Breadth of the Rectangle :&quot;); int b=sc.nextInt();
int area=l*b;
System.out.println(&quot;\nArea of Rectangle is = &quot; +area);
}
}
1
//Main
import MyPackage1.*; import MyPackage2.*; import MyPackage3.*; import java.io.*;
import
java.util.*; class Main
{
public static void main(String args[])
{
Factorial F=new Factorial(); MaxNum MN=new MaxNum(); Rectangle R=new
Rectangle();
F.Factorial();
MN.MaxNum();
R.RectArea();
}
}
Code:
Package1
package Package1;

```

```

import java.util.Scanner;
public class Factorial {
public Factorial() {
    System.out.println("Enter Number To Find Factorial:");
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int f = 1;
    for (int i = n; i > 0; i--) {
        f = f * i;
    }
    System.out.println("\nFactorial of " + n + " is: " + f); }
    public void Factorial() {
    }
}

```

Package2

```

package Package2;
import java.util.Scanner;

```

```

public class MaxNum {
    public void MaxNum() {
        System.out.println("\n1st number : ");
        Scanner sc = new Scanner(System.in);
        int mn1 = sc.nextInt();
        System.out.println("2nd number : ");
        int mn2 = sc.nextInt();
        if (mn1 >= mn2) {
            System.out.println("\nMax Num is 1st Number");
        } else {
            System.out.println("\nMax Num is 2nd Number");
        }
    }
}

```

Package3

```

package Package3;
import java.util.Scanner;
public class Rectangle {
    public void RectArea() {
        System.out.println("\nEnter Length of the Rectangle :");
        Scanner sc = new Scanner(System.in);
        int l = sc.nextInt();
        System.out.println("Enter Breadth of the Rectangle :");
        int b = sc.nextInt();
        int area = l * b;
        System.out.println("\nArea of Rectangle is = " + area);
    }
}

```

Main.java

```

package fymca_59;
import Package1.*;
import Package2.*;
import Package3.*;

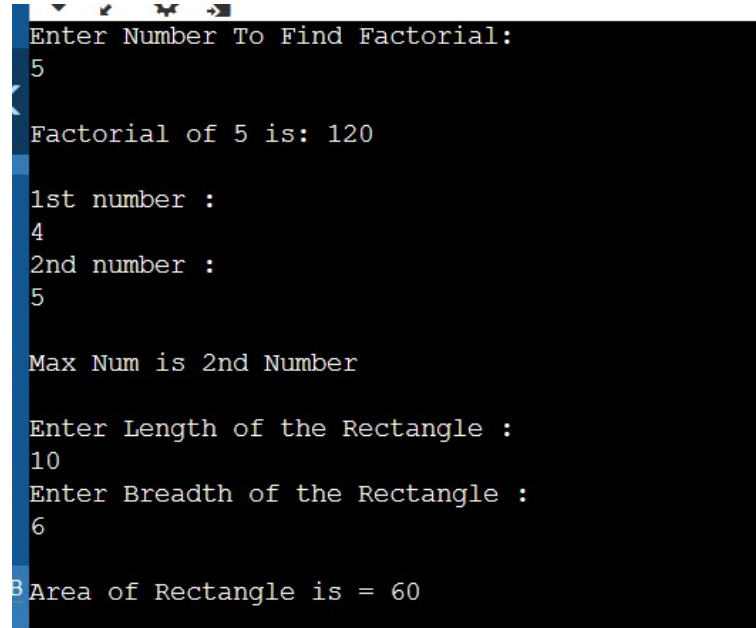
```

```

public class Main {
    public static void main(String args[]) {
        Factorial F = new Factorial();
        MaxNum MN = new MaxNum();
        Rectangle R = new Rectangle();
        F.Factorial();
        MN.MaxNum();
        R.RectArea();
    }
}

```

Output:



```

Enter Number To Find Factorial:
5
Factorial of 5 is: 120

1st number :
4
2nd number :
5
Max Num is 2nd Number

Enter Length of the Rectangle :
10
Enter Breadth of the Rectangle :
6
Area of Rectangle is = 60

```

21. Execute the given code, Identify the error in the output. Implement "this" keyword wherever necessary and describe the difference in the output.

```

class Student{
    int rollno; String name; float fee;
    Student(int rollno,String name,float fee){ rollno=rollno;
    name=name; fee=fee;
    }
    void display(){System.out.println(rollno+" "+name+" "+fee);}
}
class TestThis1{
    public static void main(String args[]){ Student s1=new Student(111,"ankit",5000f);
    Student s2=new Student(112,"sumit",6000f); s1.display();
    s2.display();
    }}

```

Code:

```

package fymca_59;
class Student {
    int rollno;

```

```

String name;
float fee;
Student(int rollno, String name, float fee) {
    this.rollno = rollno;
    this.name = name;
    this.fee = fee;
}
void display() {
    System.out.println(rollno + " " + name + " " + fee);
}
}

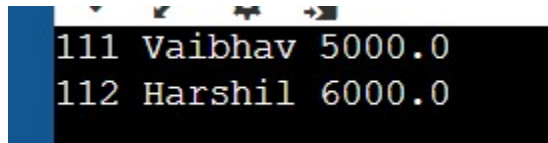
```

```

class Main {
    public static void main(String args[]) {
        Student s1 = new Student(111, "ankit", 5000f);
        Student s2 = new Student(112, "sumit", 6000f);
        s1.display();
        s2.display();
    }
}

```

Output:



```

111 Vaibhav 5000.0
112 Harshil 6000.0

```

22. Derive a scenario which has a presence of all levels of Inheritance to demonstrate your understanding about the concept. (Note - each student will work on exclusive case study)

Code:

```

package fymca_59;
//Base class - Vehicle
class Vehicle {
    String make;
    String model;
    int year;
    String color;
    public Vehicle(String make, String model, int year, String color)
    { this.make = make;
      this.model = model;
      this.year = year;
      this.color = color;
    }
    public void drive() {
        System.out.println("Vehicle is in motion.");
    }
    public void stop() {
        System.out.println("Vehicle has stopped.");
    }
}
class Car extends Vehicle {
    int numDoors;
}

```

```

    String transmissionType;
    public Car(String make, String model, int year, String color, int numDoors,
String transmissionType) {
        super(make, model, year, color);
        this.numDoors = numDoors;
        this.transmissionType = transmissionType;
    }
    public void accelerate() {
        System.out.println("Car is accelerating.");
    }
    public void brake() {
        System.out.println("Car is braking.");
    }
}

class ElectricCar extends Car {
    int batteryCapacity;
    int chargingTime;
    public ElectricCar(String make, String model, int year, String color, int
numDoors, String transmissionType,
    int batteryCapacity, int chargingTime) {
        super(make, model, year, color, numDoors, transmissionType);
        this.batteryCapacity = batteryCapacity;
        this.chargingTime = chargingTime;
    }
    public void chargeBattery() {
        System.out.println("Battery is charging.");
    }
    public void checkRange() {
        System.out.println("Checking electric range.");
    }
}

class Motorcycle extends Vehicle {
    int engineSize;
    int topSpeed;
    public Motorcycle(String make, String model, int year, String color, int
engineSize, int topSpeed) {
        super(make, model, year, color);
        this.engineSize = engineSize;
        this.topSpeed = topSpeed;
    }
    public void wheelie() {
        System.out.println("Performing a wheelie.");
    }
}

class Truck extends Vehicle {
    int cargoCapacity;
    String truckType;
    public Truck(String make, String model, int year, String color, int
cargoCapacity, String truckType) {
        super(make, model, year, color);
        this.cargoCapacity = cargoCapacity;
        this.truckType = truckType;
    }
    public void loadCargo() {
        System.out.println("Loading cargo onto the truck.");
    }
}

```



```

    }
    public void unloadCargo() {
        System.out.println("Unloading cargo from the truck.");
    }
}
public class Main {
    public static void main(String[] args) {
        Car car = new Car("Toyota", "Camry", 2022, "Red", 4, "Automatic"); ElectricCar
electricCar = new ElectricCar("Tesla", "Model S", 2022, "Blue", 4, "Automatic",
100, 8);
        Motorcycle motorcycle = new Motorcycle("Harley-Davidson", "Sportster", 2022,
"Black", 1200, 150);
        Truck truck = new Truck("Ford", "F-150", 2022, "White", 5000, "Pickup");

        car.drive();
        car.stop();
        car.accelerate();
        car.brake();

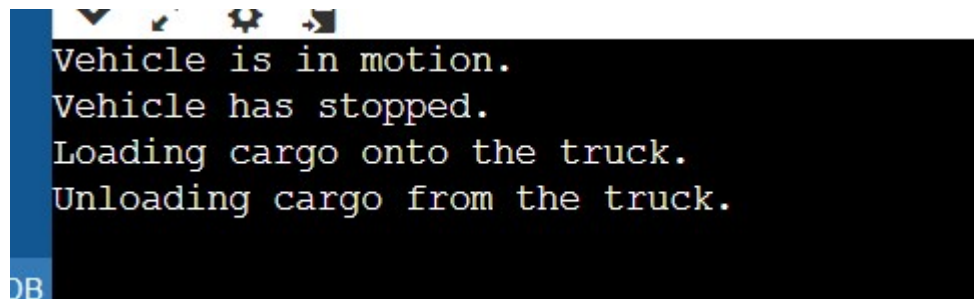
        electricCar.drive();
        electricCar.stop();
        electricCar.chargeBattery();
        electricCar.checkRange();

        motorcycle.drive();
        motorcycle.stop();
        motorcycle.wheelie();

        truck.drive();
        truck.stop();
        truck.loadCargo();
        truck.unloadCargo();
    }
}

```

Output:



```

Vehicle is in motion.
Vehicle has stopped.
Loading cargo onto the truck.
Unloading cargo from the truck.

```

By creating this scenario, we demonstrate the concept of all levels of inheritance: 1. Single Inheritance: Car class inherits from the Vehicle class, inheriting its attributes and methods while adding specific attributes and methods for cars.

2. Multilevel Inheritance: ElectricCar class inherits from the Car class, which in turn inherits from the Vehicle class. This represents multiple levels of inheritance.

3. Hierarchical Inheritance: Motorcycle class and Truck class inherit directly from the Vehicle class. They share common attributes and methods from the Vehicle class but have their own

specific attributes and methods, representing hierarchical inheritance.

23. Write a Package MCA which has one class Student. Accept student details through parameterized constructor. Write display () method to display details. Create a main class which will use package and calculate total marks and percentage.

Code:

```
package MCA;

public class Student {
    private String
name;
    private int
rollNumber;
    private int[]
marks;

    // Parameterized
constructor to accept
student details
    public
Student(String name,
int rollNumber, int[]
marks) {
        this.name =
name;

        this.rollNumber =
rollNumber;
        this.marks =
marks;
    }

    // Method to
display student
details
    public void
display() {

        System.out.println("Na
me: " + name);

        System.out.println("Ro
ll Number: " +
rollNumber);

        System.out.println("Ma
rks in each
subject:");
        for (int i =
0; i < marks.length;
i++) {

            System.out.println("Su
bject " + (i + 1) + ":
" + marks[i]);
        }
    }
}
```

```

        // Method to
        calculate total marks
        public int
        calculateTotalMarks()
        {
            int total = 0;
            for (int mark
: marks) {
                total +=
mark;
            }
            return total;
        }

        // Method to
        calculate percentage
        public float
        calculatePercentage()
        {
            int totalMarks
=
calculateTotalMarks();
            return (float)
totalMarks /
marks.length;
        }
    }

package first_project;
import MCA.Student;
public class Main_student {
    public static void main(String[] args) {
        // Creating objects of Student class and passing details
        Student student1 = new Student("Vivek", 101, new int[]{85, 90, 80, 75});
        Student student2 = new Student("Vaibhav", 102, new int[]{75, 70, 65, 60});

        // Displaying student details
        System.out.println("Student 1 details:");
        student1.display();
        System.out.println("\nStudent 2 details:");
        student2.display();

        // Calculating and displaying total marks and percentage
        System.out.println("\nStudent 1 total marks: " +
student1.calculateTotalMarks());
        System.out.println("Student 1 percentage: " +
student1.calculatePercentage() + "%");

        System.out.println("\nStudent 2 total marks: " +
student2.calculateTotalMarks());
        System.out.println("Student 2 percentage: " +
student2.calculatePercentage() + "%");
    }
}

```

Output:

```
Student 1 details:
Name: Vivek
Roll Number: 101
Marks in each subject:
Subject 1: 85
Subject 2: 90
Subject 3: 80
Subject 4: 75

Student 2 details:
Name: Vaibhav
Roll Number: 102
Marks in each subject:
Subject 1: 75
Subject 2: 70
Subject 3: 65
Subject 4: 60

Student 1 total marks: 330
Student 1 percentage: 82.5%

Student 2 total marks: 270
Student 2 percentage: 67.5%
```

24. Write a program to create a user defined package in Java.

Code:

Package fymca_59 and class Main.

```
package first_project;
```

```
public class Main1 { public void show()
{
System.out.println("Hello, it's Vivek, How are you?"); } public static void
main(String args[])
{
Main1 obj = new Main1();
obj.show();
}
}
```

Package MCA and Class MyClass.

```
package MCA;
```

```
import first_project.Main1;
```

```
public class MyClass {  
    public static void main(String args[])  
    {  
        Main1 obj = new Main1(); obj.show();  
    }  
}
```

Output:

```
Hello, it's Vivek, How are you?
```

25. Write a class Dept with a final keyword as class. Assign the class value as “SYMCA”. Restrict object of Dept from overwriting the value of the class to any other value as “TYMCA” or “FYMCA.”

Code:

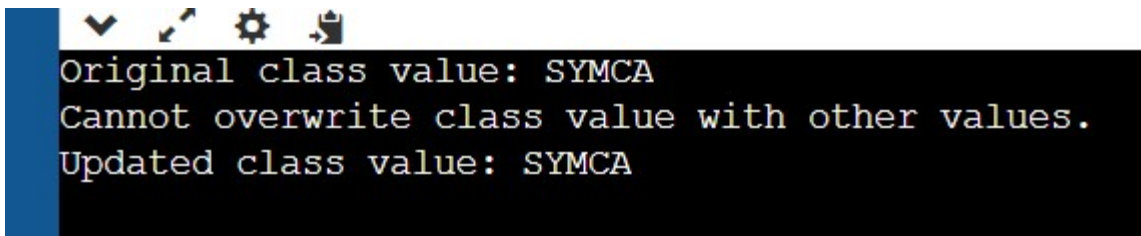
```
package fymca_59;  
public final class Deptt {  
    private static final String CLASS_VALUE = "SYMCA";
```

```

public static String getClassValue() {
return CLASS_VALUE;
}
private Deptt() {
throw new AssertionError("Cannot instantiate a final class."); }
public static void setClassValue(String value) {
if (!"SYMCA".equals(value)) {
System.out.println("Cannot overwrite class value with other values."); }
}
public static void main(String[] args) {
System.out.println("Original class value: " + Deptt.getClassValue());
Deptt.setClassValue("TYMCA");
System.out.println("Updated class value: " + Deptt.getClassValue()); }
}

```

Output:



```

Original class value: SYMCA
Cannot overwrite class value with other values.
Updated class value: SYMCA

```

26. Implement class parent inheriting class child 1. In a simple way extend class child 1 to child 2. Write the same function name in all the classes as showdata and write a same variable name "name" using super keyword show the call of parent class function or variable from the respective child class.

Code:

```

package fymca_59;
class Parent {
String name;
public Parent(String name) {
this.name = name;
}
public void showData() {
System.out.println("Parent Name: " + name);
}
}
class Child1 extends Parent {
public Child1(String name) {
super(name);
}

public void showData() {
super.showData();
System.out.println("Child1 Name: " + name);
}
}
class Child2 extends Child1 {
public Child2(String name) {

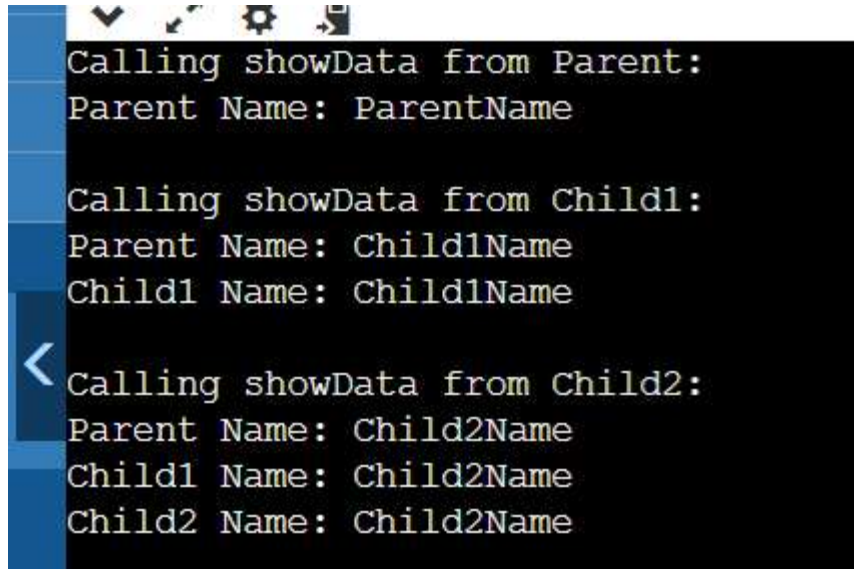
```

```

    super(name);
}
public void showData() {
    super.showData();
    System.out.println("Child2 Name: " + name);
}
}
public class twenty-six {
    public static void main(String[] args) {
        Parent parent = new Parent("ParentName");
        Child1 child1 = new Child1("Child1Name");
        Child2 child2 = new Child2("Child2Name");
        System.out.println("Calling showData from Parent:");
        parent.showData();
        System.out.println();
        System.out.println("Calling showData from Child1:");
        child1.showData();
        System.out.println();
        System.out.println("Calling showData from Child2:");
        child2.showData();
    }
}

```

Output:



```

Calling showData from Parent:
Parent Name: ParentName

Calling showData from Child1:
Parent Name: Child1Name
Child1 Name: Child1Name

< Calling showData from Child2:
Parent Name: Child2Name
Child1 Name: Child2Name
Child2 Name: Child2Name

```

27. Demonstrate wrapper class for Integer class using all function

Code:

```

package fymca_59;
class MyInteger {
    private Integer value;
    public MyInteger(Integer value) {
        this.value = value;
    }
    public Integer getValue() {
        return value;
    }
    public void setValue(Integer value) {
        this.value = value;
    }
}

```

```

public String toString() {

return value.toString();
}
public int hashCode() {
return value.hashCode();
}
public boolean equals(Object obj) {
if (this == obj) {
return true;
}
if (!(obj instanceof MyInteger)) {
return false;
}
MyInteger other = (MyInteger) obj;
return this.value.equals(other.value);
}
public static MyInteger parseInt(String s) throws NumberFormatException {
return new MyInteger(Integer.parseInt(s));
}
public byte byteValue() {
return value.byteValue();
}
public short shortValue() {
return value.shortValue();
}
public int intValue() {
return value.intValue();
}
public long longValue() {
return value.longValue();
}
public float floatValue() {
return value.floatValue();
}
public double doubleValue() {
return value.doubleValue();
}
public String toOctalString() {
return Integer.toOctalString(value);
}
public String toHexString() {
return Integer.toHexString(value);
}
public String toBinaryString() {
return Integer.toBinaryString(value);
}
}
public class Wrapper {
public static void main(String[] args) {
// Creating a MyInteger object
MyInteger myInt = new MyInteger(59);
System.out.println("Value: " + myInt.getValue());
System.out.println("String representation: " + myInt.toString());
System.out.println("Hash code: " + myInt.hashCode());
}
}

```



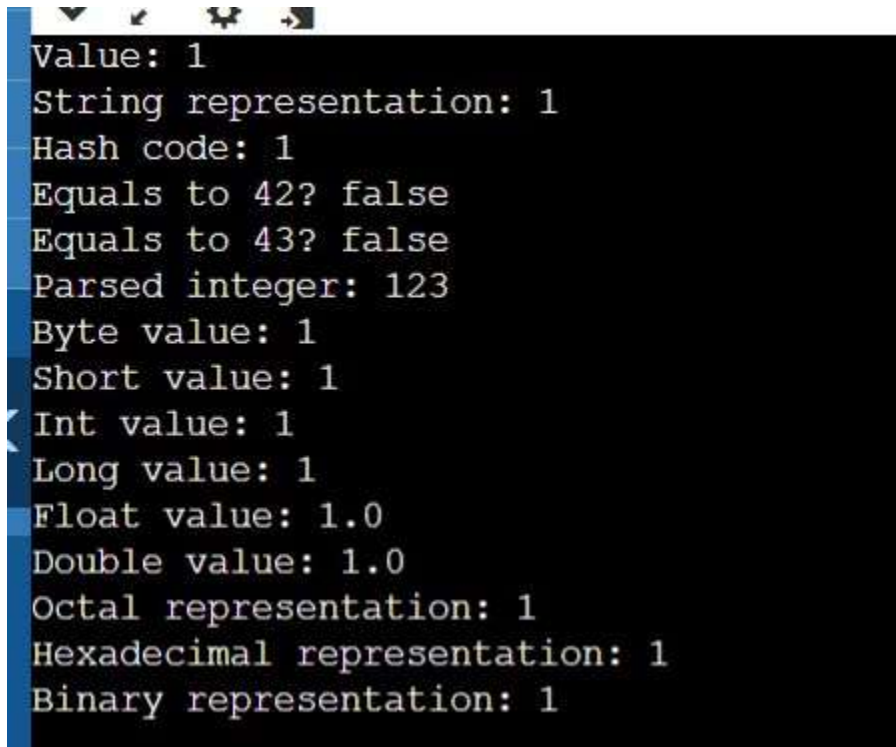
```

System.out.println("Equals to 42? " + myInt.equals(new MyInteger(42)));
System.out.println("Equals to 43? " + myInt.equals(new MyInteger(43)));

MyInteger parsedInt = MyInteger.parseInt("123");
System.out.println("Parsed integer: " + parsedInt.getValue());
System.out.println("Byte value: " + myInt.byteValue());
System.out.println("Short value: " + myInt.shortValue());
System.out.println("Int value: " + myInt.intValue());
System.out.println("Long value: " + myInt.longValue());
System.out.println("Float value: " + myInt.floatValue());
System.out.println("Double value: " + myInt.doubleValue());
System.out.println("Octal representation: " + myInt.toOctalString());
System.out.println("Hexadecimal representation: " + myInt.toHexString());
System.out.println("Binary representation: " + myInt.toBinaryString()); }
}

```

Output:



```

Value: 1
String representation: 1
Hash code: 1
Equals to 42? false
Equals to 43? false
Parsed integer: 123
Byte value: 1
Short value: 1
Int value: 1
Long value: 1
Float value: 1.0
Double value: 1.0
Octal representation: 1
Hexadecimal representation: 1
Binary representation: 1

```

28. Write a program for writing book with author class with different type of books as per domain. Demonstrate the multilevel Inheritance. Demonstrate Super and Final keyword with appropriate assumed data and functions.

Code:

```

package fymca_59;
class Author {
private String name;
public Author(String name) {
this.name = name;
}
public String getName() {
return name;
}
}

```

```

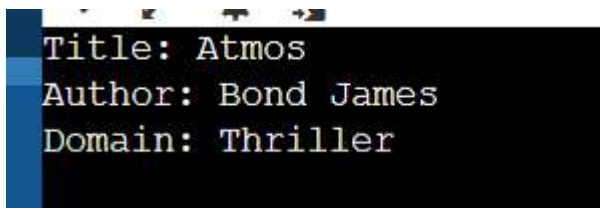
}
class BaseBook {
    private String title;
    private Author author;
    public BaseBook(String title, Author author) {
        this.title = title;
        this.author = author;
    }
    public String getTitle() {
        return title;
    }
    public Author getAuthor() {

return author;
    }
}
class DomainBook extends BaseBook {
    private final String domain;
    public DomainBook(String title, Author author, String domain) {
super(title, author);
        this.domain = domain;
    }
    public String getDomain() {
        return domain;
    }
    public final void displayInfo() {
        System.out.println("Title: " + getTitle());
        System.out.println("Author: " + getAuthor().getName());
        System.out.println("Domain: " + domain);
    }
}
public class Main {
    public static void main(String[] args) {
        Author author = new Author("James Clear");
        DomainBook book = new DomainBook("Atomic Habits", author, "Psychology");

        book.displayInfo();
    }
}

```

Output:



```

Title: Atmos
Author: Bond James
Domain: Thriller

```

29. Create an interface for the class Subject. Display the marks of the subject derived from practical and theory class.

Code:

```

package fymca_59;
interface Subject {

```

```

    void displayMarks();
}
class Practical implements Subject {
    private int practicalMarks;

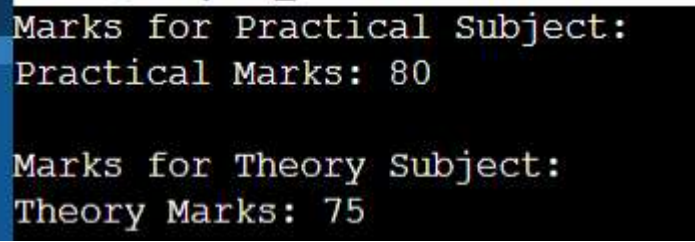
    public Practical(int practicalMarks) {
        this.practicalMarks = practicalMarks;
    }

    public void displayMarks() {
        System.out.println("Practical Marks: " + practicalMarks);
    }
}
class Theory implements Subject {

    private int theoryMarks;
    public Theory(int theoryMarks) {
        this.theoryMarks = theoryMarks;
    }
    public void displayMarks() {
        System.out.println("Theory Marks: " + theoryMarks);
    }
}
public class Main {
    public static void main(String[] args) {
        Practical practical = new Practical(80);
        Theory theory = new Theory(75);
        System.out.println("Marks for Practical Subject:");
        practical.displayMarks();
        System.out.println("\nMarks for Theory Subject:");
        theory.displayMarks();
    }
}

```

Output:



```

Marks for Practical Subject:
Practical Marks: 80

Marks for Theory Subject:
Theory Marks: 75

```

30. Write a student admission process like student information, Roll no. allocation process, Exam process(To take marks from the user and display marks) in a package "Student". Create the object for MCA and Comp. Engg student by importing package using fully qualified way.

Code:

Main.java

```

package Student;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {

```

```

Scanner scanner = new Scanner(System.in);
System.out.print("Enter MCA student's name: ");
String mcaName = scanner.nextLine();
System.out.print("Enter MCA student's age: ");
int mcaAge = scanner.nextInt();
StudentInfo mcaStudent = new StudentInfo(mcaName, mcaAge); int mcaRollNumber =
RollNumberAllocation.allocateRollNumber(); System.out.println("MCA Student -
Name: " + mcaStudent.getName() + ", Age: " + mcaStudent.getAge() + ", Roll Number:
" + mcaRollNumber);
ExamProcess.conductExam(mcaStudent);
System.out.print("Enter Computer Engineering student's name: ");
scanner.nextLine();
String compEnggName = scanner.nextLine();

System.out.print("Enter Computer Engineering student's age: "); int
compEnggAge = scanner.nextInt();
StudentInfo compEnggStudent = new StudentInfo(compEnggName, compEnggAge); int
compEnggRollNumber = RollNumberAllocation.allocateRollNumber();
System.out.println("Computer Engineering Student - Name: " +
compEnggStudent.getName() + ", Age: " + compEnggStudent.getAge() + ", Roll Number:
" + compEnggRollNumber);
ExamProcess.conductExam(compEnggStudent);
}
}

```

StudentInfo.java

```

package Student;
public class StudentInfo {
    private String name;
    private int age;
    public StudentInfo(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
}

```

RollNumberAllocation.java

```

package Student;
public class RollNumberAllocation {
    private static int rollNumberCounter = 1059;
    public static int allocateRollNumber() {
        return rollNumberCounter++;
    }
}

```

ExamProcess.java

```

package Student;

```

```

import java.util.Scanner;
public class ExamProcess {
    public static void conductExam(StudentInfo student) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter marks for " + student.getName() + ": "); int
marks = scanner.nextInt();
        System.out.println("Marks obtained by " + student.getName() + ": " + marks);    }
}

```

Output:

31. Create an abstract class 'Bank' with an abstract method 'getBalance'. \$100, \$150 and \$200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

Code:

```

package fymca_59;
abstract class Bank {
    abstract int getBalance();
}
class BankA extends Bank {
    private int balance = 0;

    BankA(int amount) {
        balance += amount;
    }
    @Override
    int getBalance() {
        return balance;
    }
}
class BankB extends Bank {
    private int balance = 0;

    BankB(int amount) {
        balance += amount;
    }
    @Override
    int getBalance() {
        return balance;
    }
}

```

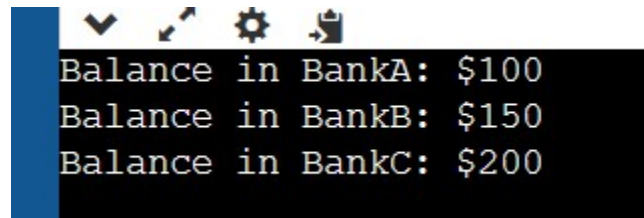
```

}
class BankC extends Bank {
    private int balance = 0;
    BankC(int amount) {
        balance += amount;
    }
    @Override
    int getBalance() {
        return balance;
    }
}
public class Main {
    public static void main(String[] args) {
        // Creating objects for each bank
        BankA bankA = new BankA(100);
        BankB bankB = new BankB(150);
        BankC bankC = new BankC(200);

        // Calling getBalance method for each bank
        System.out.println("Balance in BankA: $" + bankA.getBalance());
        System.out.println("Balance in BankB: $" + bankB.getBalance());
        System.out.println("Balance in BankC: $" + bankC.getBalance());
    }
}

```

Output:



```

Balance in BankA: $100
Balance in BankB: $150
Balance in BankC: $200

```

32. Declare a protected integer attribute called legs, which records the number of legs for this animal.

Code:

```

package fymca_59;
abstract class Animal {
    protected int legs;
    Animal(int legs) {
        this.legs = legs;
    }
    abstract void makeSound();
}
class Dog extends Animal {
    Dog() {
        super(4);
    }
    @Override
    void makeSound() {
        System.out.println("Woof");
    }
}

```

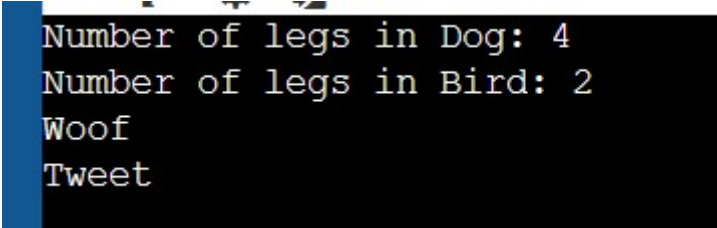
```

class Bird extends Animal {
    Bird() {
        super(2);
    }
    @Override
    void makeSound() {

        System.out.println("Tweet");
    }
}
public class Main {
    public static void main(String[] args) {
        Dog dog = new Dog();
        Bird bird = new Bird();
        System.out.println("Number of legs in Dog: " + dog.legs);
        System.out.println("Number of legs in Bird: " + bird.legs);
        dog.makeSound();
        bird.makeSound();
    }
}

```

Output:



```

Number of legs in Dog: 4
Number of legs in Bird: 2
Woof
Tweet

```

Observation:

1. Basic java programs was performed in Eclipse IDE using various methods of java.
2. Scanner was used to take input from user.
3. Classes, Instance variables and many more methods are used.
4. User defined package which helps in organizing classes into namespaces.
5. Interfaces declare methods that must be implemented by classes that implement the interface.
6. Inheritance allows a class to inherit properties and behaviors from another class.
7. Polymorphism allows objects to be treated as instances of their parent class.
8. Abstract classes cannot be instantiated and may contain abstract methods and Wrapper classes provide a way to use primitive data types as objects.