Experiment No.1.3

**Date:** 02-02-2024

**Aim:** Fundamentals of Java Programming

**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

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

public class one {

public static void main(String[] args) {

System.out.print("Hello World!");

}

}

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.

package Lab1\_3;

import java.util.\*;

public class shoppingcart {

private String productName;

private int productId;

private Date purchaseDate;

private String productCategory;

public shoppingcart(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<shoppingcart> cartList) {

return cartList.size();

}

public static Map<Date, Integer> getFinalValuesOfEachDayPurchase(List<shoppingcart> cartList) {

Map<Date, Integer> finalValues = new HashMap<>();

for (shoppingcart 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) {

shoppingcart cart1 = new shoppingcart("Laptop", 1001, new Date(), "Electronics");

shoppingcart cart2 = new shoppingcart("Mobile Phone", 1002, new Date(), "Electronics");

shoppingcart cart3 = new shoppingcart("Dress", 2001, new Date(), "Clothing");

shoppingcart cart4 = new shoppingcart("Shoes", 2002, new Date(), "Footwear");

shoppingcart cart5 = new shoppingcart("Earphone", 2000, new Date(), "Electronics");

List<shoppingcart> 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 (shoppingcart 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);

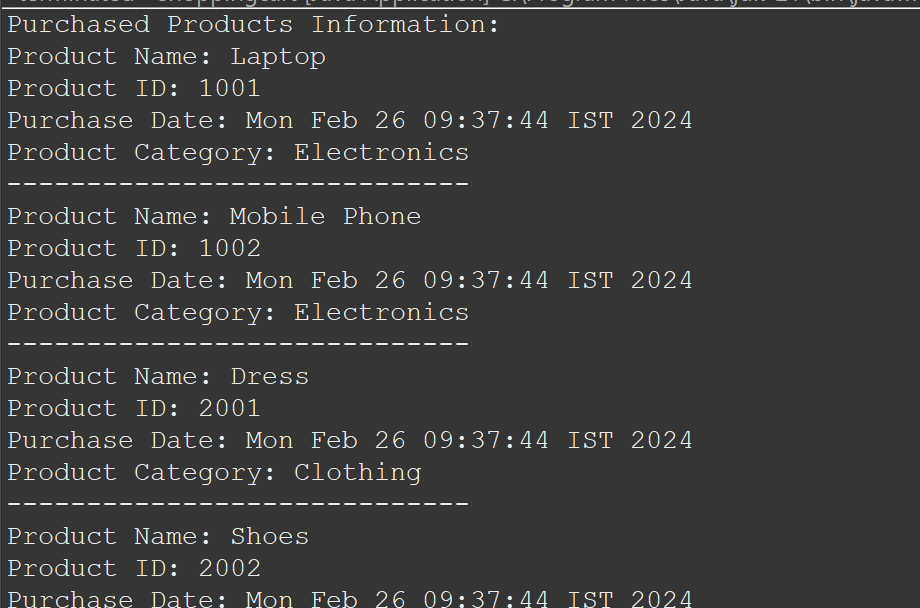
Map<Date, Integer> finalValues = *getFinalValuesOfEachDayPurchase*(cartList);

System.***out***.println("Final values of each day purchase: " + finalValues);

}

}

Output:



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.

package Lab1\_3;

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(); // Consume newline character after nextDouble()

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);

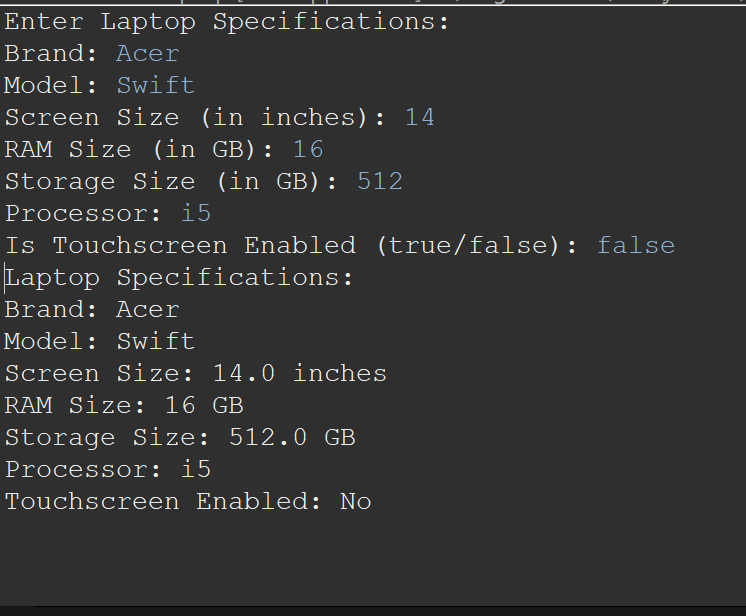
laptop.displaySpecifications();

scanner.close();

}

}

Output:



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.

package Lab1\_3;

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[]:");

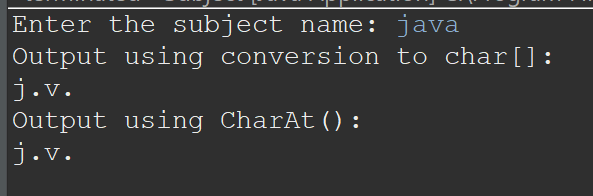
System.***out***.println(charArray[0] + "." + charArray[charArray.length - 2] + ".");

System.***out***.println("Output using CharAt():");

System.***out***.println(subjectName.charAt(0) + "." + subjectName.charAt(subjectName.length() - 2) + ".");

}

}



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

Programmer "Is-A" relationship.

package Lab1\_3;

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);

}

}

Main1.class

package Lab1\_3;

public class Main1{

public static void main(String[] args) {

Employee employee = new Employee("Vivek", 1001, 50000);

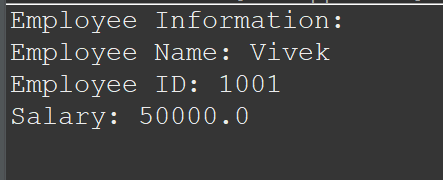
System.***out***.println("Employee Information:");

employee.displayEmployeeInfo();

System.***out***.println();

}

}



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

7. W.A.P based on the concept of Inheritance using Shape class of various geometrical

figures.Calculate Area for different shapes

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

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);

}

}

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

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

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

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

be seen.

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

parameters for Volume.

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,&quot;blue&quot;); printCircle(myc); colorCircle(myc,&quot;black&quot;);

printCircle(myc);

}

public static void colorCircle(Circle c , String color)

1

{

c.color = color;

}

public static void printCircle(Circle c)

{

System.out.println(&quot;The Area of the Circle of the radius&quot;+c.getRadius()+&quot; is

&quot;+c.findArea()); System.out.println(&quot;The Color of the circle is: &quot;+c.color);

}

}

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

}

}

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

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

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());

}

}

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&gt;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&gt;=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();

}

}

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();

}}

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)

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.

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

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

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.

27. Demonstrate wrapper class for Integer class using all function

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.

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

practical and theory class

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.

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

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

animal.