23CSE111

OBJECT ORIENTED PROGRAMMING

LAB REPORT



Department of Computer Science Engineering Amrita School of Computing

Amrita Vishwa Vidyapeetham, Amaravati Campus

Name: N.Manikanta Gowtham

Roll No: 24231

Verified By :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Experiment | Page No | Remarks | Signature |
| 1 | Installation Process of JDK | 3-4 |  |  |
| 2 | Simple java program for printing basic details of student | 5 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

INDEX

# WEEK-1

1. Process of Installing JDK (Java Development Kit)

Installing JDK (Java Development Kit):

* 1. Download JDK:
* Go to the Oracle JDK download page in google and click on JDK-21 version which is Long term support (LTS) version.
* Click the download link as your operating system (Windows, macOS, or Linux).
  1. Install JDK:
* Once downloaded, run the installer.
* Follow the given instructions and keep clicking "Next" until it is done.
  1. Set Environment Variables (Windows):
* Open file explorer, then right click on This PC next select on properties then it will take you to the settings app then click on advanced system settings and then click on Environment Variables.
* Click on path and new under System Variables:

Variable value: The folder address where JDK is installed (like

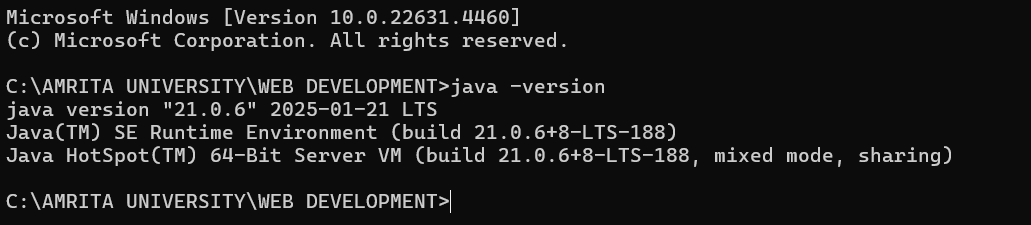
C:\Program Files\Java\jdk-21\bin)

* Find Path under System Variables, click New, and add the path of the jdk-21(C:\Program Files\Java\jdk-21\bin)



Checking JDK Version: -

* 1. Open Command Prompt:
* Presswin+R, typecmd, and press Enter.
  1. Check Version:
* Type java -version and press Enter.
* Type javac --version and press Enter.



1. Simple Java Program for printing Name, Class, Roll No, of a Student

Write your code in Notepad and execute it in cmd prompt

CODE: -

class Main

{

public static void main(String[] args)

{

System.out.println("Name:N.Manikanta Gowtham");

System.out.println("Class:CSE-C");

System.out.println("Roll No:24231);

}

}

Output: -



|  |  |  |
| --- | --- | --- |
| 1 | Syntax error | Semicolon added |
| 2 | Runtime error | Copied correct path |
| 3 | Name error | rectified |

Week-2

1)

AIM ;

Simple Java Program for finding simple interest by taking input from

User

Code:

import java.util.Scanner;

class Main {

  public static void main(String[] args) {

    Scanner input = new Scanner(System.in);

    System.out.print("Enter the principal: ");

    double principal = input.nextDouble();

    System.out.print("Enter the rate: ");

    double rate = input.nextDouble();

    System.out.print("Enter the time: ");

    double time = input.nextDouble();

    double interest = (principal \* time \* rate) / 100;

    System.out.println("Principal: " + principal);

    System.out.println("Interest Rate: " + rate);

    System.out.println("Time Duration: " + time);

    System.out.println("Simple Interest: " + interest);

    input.close();

  }

}

Output



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | rectification |
| 1 | Runtime error | Incorrect path | Copied correct path |
| 2 | Syntax error | { missing | { added |
| 3 | Logical error | Wrong formula | Formula rectified |

2)

AIM:

Write a simple program to calculate factorial of a number and read the

input from user

code:

class Test {

static int factorial(int n)

{

int res = 1, i;

for (i = 2; i <= n; i++)

res \*= i;

return res;

}

public static void main(String[] args)

{

int num = 5;

System.out.println("Factorial of " + num + " is "

+ factorial(5));

}

}

Output



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | Rectification |
| 1 | Undeclared variable error | Missing variable | Variable declared |
| 2 | Missing import statement | Not importing packages | Packages imported |
| 3 | Logical error | Wrong formula | Formula rectified |

3)

AIM:

Write a program to to calculate the fibonacii sequence and take the input from user

Code:

import java.util.\*;

class fibo

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int num;

int f3;

int f1 = 0;

int f2 = 1;

int i = 2;

System.out.print("Enter a number:");

num = sc.nextInt();

System.out.println(f1);

System.out.println(f2);

while(i<num)

{

f3 = f1+f2;

f1 = f2;

f2 = f3;

System.out.println(f3);

i = i+1;

}

}

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | Rectification |
| 1 | Logical error | Incorrect formula | Formula rectified |
| 2 | Run-time error | Incorrect path | Added correct path |
|  |  |  |  |

4)AIM;

Write a java program to convert temperature from Fahrenheit to celsius

Code:

class Geeks {

    public static void main(String[] args)

    {

        double f = 50.0, c = 0.0;

        c = (f - 32) / 1.8;

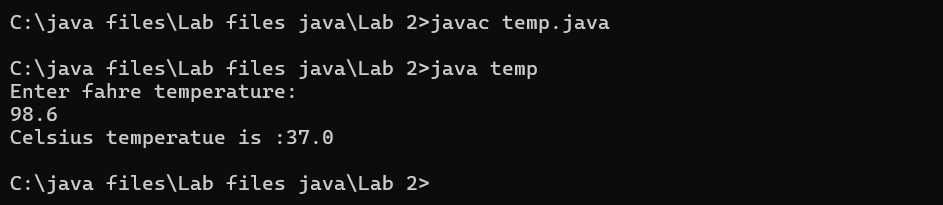
        System.out.println(

            "value of temperature in celsius:" + c);

    }

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | rectification |
| 1 | Syntax error | Missing ” | “ is added |
| 2 | Missing import error | Util package missing | Util package added |
|  |  |  |  |

5)AIM:

Write a java program to convert temperature from Celsius to Fahrenheit

Code:

class celsiustofahrenheit {

    public static void main(String[] args)

    {

        double celsius = 10.0, fahrenheit = 0.0;

        fahrenheit = (celsius \* 1.8) + 32;

        System.out.println(

            " value of temperature in fahrenheit:"

            + fahrenheit);

    }

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | Rectification |
| 1 | Runtime error | Incorrect path selection | Correct path added |
| 2 | Logical error | Incorrect logic | Correct logic |
|  |  |  |  |

6)AIM:

Write a simple program to find the area of rectangle:

Code:

import java.util.\*;

class Area{

     public static void main(String[] args) {

            int area;

           Scanner sc = new Scanner(System.in);

           System.out.println("Enter Length:");

           int l = sc.nextInt();

           System.out.println("Enter Breadth:");

           int b = sc.nextInt();

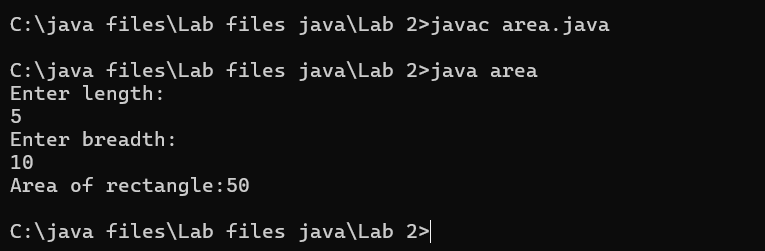
           area = l\*b;

           System.out.println("Area of Rectangle :"+area);

     }

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | Rectification |
| 1 | Syntax error | Semi colon missing | Semi colon added |
| 2 | Missing import error | Import package missing | Import package added |
|  |  |  |  |

7)AIM:

Write a program to find the area of triangle by using heron’s formula

take the input from the user

Code:

port java.io.\*;

class GFG {

    static double area(double h, double b)

    {

        return (h \* b) / 2;

    }

    public static void main(String[] args)

    {

        double h = 10;

        double b = 5;

        System.out.println("Area of the triangle: "

                           + area(h, b));

    }

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Error type | Reason for error | Rectification |
| 1 | Logical error | Incorrect formula | Formula rectified |
| 2 | Name error | Undeclared variable | Variable declared |
|  |  |  |  |

WEEK 3

Aim:

To create java program with following instructions

1.Create a class with name car

2. Create four attributes named car\_color ,Car\_brand,fuel\_type,mileage

3. Create three methods named start(), stop(). Service()

4. Create three objects named car1,car2 and car3

Code:

import java.util.\*;

class car

{

public String Car\_color;

public String Car\_brand;

public String fuel\_type;

public int mileage;

public void start()

{

System.out.println("Car Started:");

System.out.println("Car color is :"+Car\_color);

System.out.println("Car Brand is:"+Car\_brand);

System.out.println("Car fuel type is:"+fuel\_type);

System.out.println("Car mileage is:"+mileage);

}

public void service()

{

System.out.println("Car Started:");

System.out.println("Car color is :"+Car\_color);

System.out.println("Car Brand is:"+Car\_brand);

System.out.println("Car fuel type is:"+fuel\_type);

System.out.println("Car mileage is:"+mileage);

}

public void stop()

{

System.out.println("Car Started:");

System.out.println("Car color is :"+Car\_color);

System.out.println("Car Brand is:"+Car\_brand);

System.out.println("Car fuel type is:"+fuel\_type);

System.out.println("Car mileage is:"+mileage);

}

public static void main(String args[])

{ System.out.println("\n manikanta\n\n");

car car1 = new car();

car1.Car\_color = "Blue";

car1.Car\_brand = "BMW";

car1.fuel\_type = "Deisel";

car1.mileage = 10;

car1.start();

car car2 = new car();

car2.Car\_color = "Red";

car2.Car\_brand = "Tesla";

car2.fuel\_type = "EV";

car2.mileage = 300;

car2.stop();

car car3 = new car();

car3.Car\_color = "Yellow";

car3.Car\_brand = "MAHINDRA";

car3.fuel\_type = "Petrol";

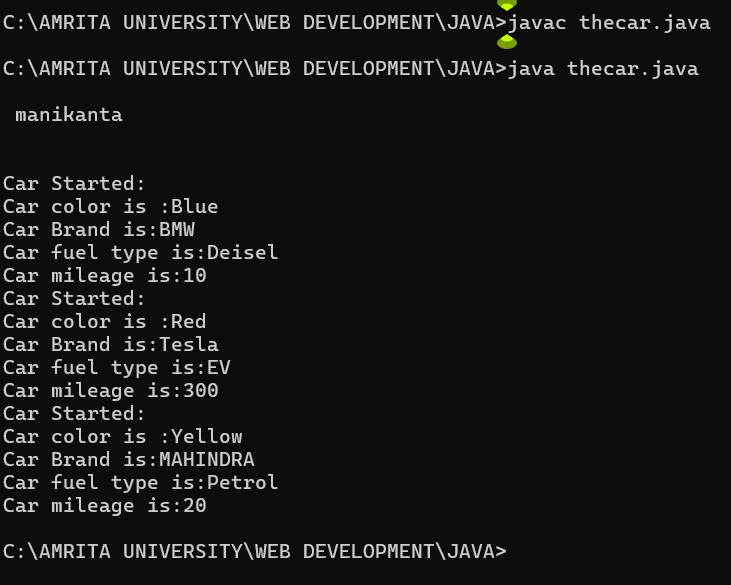
car3.mileage = 20;

car3.service();

}

}

Output:



**Class Diagram:**

|  |
| --- |
| Car |
| + car\_color: String  + car\_brand: String  + fuel\_type: String  + mileage: int |
| + Car(): void  + start(): void  + service(): void  + stop(): void |

**Concepts to be known:**

1. public String car\_color; - Used to declare a variable named car\_color, with data type as String with public accessibility.
2. Car(String car\_color,String car\_brand,String fuel\_type,int mileage){ } – It is a constructor (method with name same as class), which requires parameters such as car\_color (String data-type) and so on.
3. this.car\_color=car\_color; - “this” is a default method, which is used to point to the instance variables.
4. public void start(){} – used to declare a method, which will return nothing(void) in public accessibility.
5. Car car1=new Car("Red","Maruti","Diesel",20); - used to create a object in class Car, with object name as car1.

car1.start(); - Calling a method, under object car1

2.)

AIM:

To create a class bankAccount with methods deposit() and withdrawl

Code:

class BankAccount

{

private double balance;

public BankAccount(double initialBalance)

{

if(initialBalance > 0)

{

this.balance = initialBalance;

}

else

{

this.balance = 0;

}

}

public void deposit(double amount)

{

if(amount>0)

{

balance = balance+amount;

System.out.println("Deposited $:"+amount);

}

else

{

System.out.println("Deposited amount must be positive");

}

}

public double getBalance()

{

return balance;

}

}

public class Main1

{

public static void main(String args[])

{

BankAccount account = new BankAccount(1000);

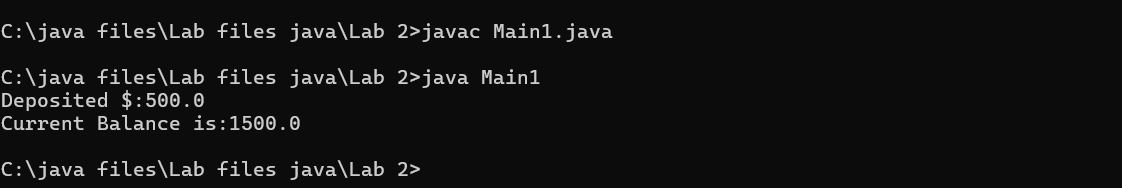
account.deposit(500);

System.out.println("Current Balance is:"+account.getBalance());

}

}

Output:

****

**Class Diagram:**

|  |
| --- |
| BankAccount |
| * name: String * Accno: int * CurrBal: int |
| BankAccount: void  + withdraw(int WAmt): void  + deposit(int DAmt): int |

**Concepts to be known:**

1. private String name; - Used to declare a variable named name, with data type as String with private accessibility.
2. BankAccount(String name,int Accno,int CurrBal){ } – It is a constructor (method with name same as class), which requires parameters such as name (String data-type) and so on.
3. this.CurrBal=CurrBal; - “this” is a default method, which is used to point to the instance variables.
4. public void withdraw(int WAmt){ } – used to declare a method, which will return nothing(void) in public accessibility, which requires a parameter WAmt(integer data type).
5. public int deposit(int DAmt){} - used to declare a method, which will return integer data type in public accessibility, which requires a parameter DAmt(integer data type).
6. BankAccount cust1=new BankAccount("Ram",5587,20000); - used to create a object in class BankAccount, with object name as cust1.
7. cust1.withdraw(50000); - Calling a method, under object cust1, by passing a parameter.

System.out.println("Your current balance after depositing money is:"+cust1.deposit(25000)); - Deposit method will return the value, which will be directly printed.

WEEK-4

**1.AIM:**

**WRITE A JAVA PROGRAM WITH CLASS NAMED “Book”. THE CLASS SHOUKD CONTAIN VARIOUS ATTRIBUTES SUCH AS TITLE, AUTHOR, YEAR OF**

**PUBLICATION. IT SHOULD ALSO CONTAIN A CONSTRUCTOR WITH**

**PARAMETERS WHICH INITIALIZES TITLE, AUTHOR, YEAR OF PUBLICATION**

**AND CREATE A METHOD WHICH DISPLAYS THE DETAILS OF 2 BOOKS.**

**PROGRAM:**

**public class Book {**

**public String title;**

**public String author;**

**public int year;**

**Book(String title, String author, int year) {**

**this.title = title;**

**this.author = author;**

**this.year = year;**

**}**

**public void displayDetails() {**

**System.out.println("Title: " +title);**

**System.out.println("Author: " +author);**

**System.out.println("Year of Publication" +year);**

**}**

**public static void main(String[] args) {**

**Book b1 = new Book("Math", "Ramanujan", 1950);**

**Book b2 = new Book("Physics", "CV Raman", 1960);**

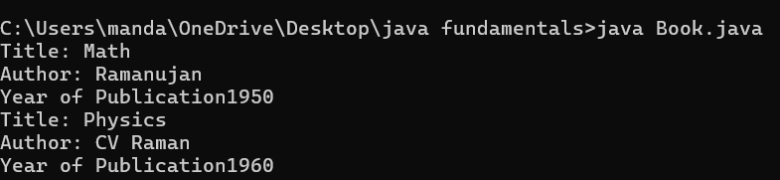
**b1.displayDetails();**

**b2.displayDetails();**

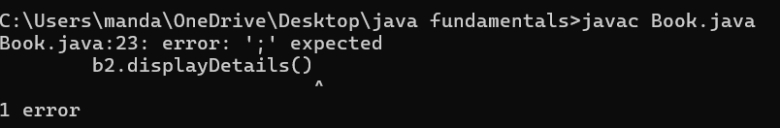
**}**

**}**

**Output:**



**NEGATIVE CASE:**



**ERROR:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **ERROR TYPE** | **Reason for error** | **Rectification** |
| **1.** | Syntax error | No semicolon | Semicolon added |
| **2.** | Runtime error | Incorrect path | Copied correct path |

**CLASS DIAGRAM:**

|  |
| --- |
| Book |
| -title: String  -author: String  -year: int |
| + Book(title: String, author:String, year: int) + displayDetails(): void |

**IMPORTANT POINTS:**

1. **Constructor**:

* The constructor Book(String, String, int) is used to initialize the object when it is created.
* The keyword **this** is used to differentiate between class attributes and constructor parameters.

2.**Method**:

* The method displayDetails() is used to display the book details.
* The **System.out.println()** method prints the details to the console.

3. **Object Creation**:

* Two objects b1 and b2 are created using the constructor.

**2.AIM:**

**WRITE A JAVA PROGRAM WITH CLASS NAMED “MyClass” WITH A STATIC VARIABLE COUNT OF INT TYPE. INTIALIZE IT TO ZERO AND A CONSTANT VARIABLE “Pi” OF TYPE DOUBLE INITIALIZED TO “3.14” AS ATTRIBUTES OF THAT CLASS. NOW DEFINE A CONSTRUCTOR FOR “MyClass”, THAT INCREMENTS THE COUNT VARIABLE EACH TIME AN OBJECT OF “MyClass”**

**IS CREATED. FINALLY, PRINT THE FINAL VALUES OF ‘COUNT’ AND ‘PI’ VARIABLES AND**

**CREATE 3 OBJECTS.**

**PROGRAM:**

public class MyClass {

static int count = 0;

static final double pi = 3.14;

MyClass() {

count++;

}

public static void main(String[] args) {

MyClass obj1 = new MyClass();

MyClass obj2 = new MyClass();

MyClass obj3 = new MyClass();

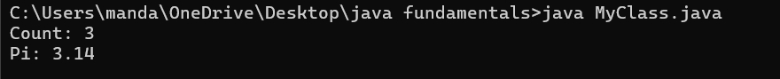
System.out.println("Count: " +count);

System.out.println("Pi: " +pi);

}

}

**OUTPUT:**



**NEGATIVE CASE:**



**ERROR:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error Type** | **Reason for error** | **Rectification** |
| **1.** | No class | No class name declared | Created class named ‘MyClass’ |
| **2.** | Syntax error | Not added keyword | Added keyword named ‘new’ |

**CLASS DIAGRAM:**

|  |
| --- |
| MyClass |
| -count: int (static)  -pi: double (static, final) |
| +MyClass()  +main(args: String[]):void |

**IMPORTANT POINTS:**

**1.Static Keyword**

* Static members belong to the **class, not to individual objects**.
* Only one copy of the static variable is maintained for all objects.

**2.Static Variable**

* **static int count**:
  + Shared among all objects of the class.
  + It is initialized only once and not for every object.
  + It increments every time the constructor is called.

**3.Final Variable**

* **static final double pi**:
  + The **final** keyword makes the variable constant.
  + Its value **cannot be changed** once assigned.
  + It must be initialized at the time of declaration.