All Code Files

Folder: .

File: generate_code_pdf.py

```
import os
from fpdf import FPDF

# Configuration
OUTPUT_PDF = "All_Code.pdf"
CODE_EXTENSIONS = [".java", ".py", ".c", ".cpp", ".js", ".ts", ".cs", ".rb", ".go",
".php", ".swift", ".kt", ".scala"]

class PDF(FPDF):
    def __init__(self):
        super().__init__()
        self.set_auto_page_break(auto=True, margin=15)
        self.add_page()
        self.set_font("Courier", size=10)

    def header(self):
        self.set_font("Arial", 'B', 14)
        self.cell(0, 10, "All Code Files", ln=True, align="C")
        self.ln(5)
        self.set_font("Courier", size=10)

    def chapter_title(self, title):
        self.set_font("Arial", 'B', 12)
        self.set_fill_color(220, 220, 220)
        self.cell(0, 8, title, ln=True, fill=True)
        self.ln(2)
        self.set_font("Courier", size=10)

    def code_block(self, code):
        self.set_font("Courier", size=10)
        for line in code.splitlines():
            self.multi_cell(0, 5, line)
        self.ln(2)

def collect_code_files(root_dir):
    code_files = []
    for folder, _, files in os.walk(root_dir):
        for file in files:
            if any(file.endswith(ext) for ext in CODE_EXTENSIONS):
                rel_dir = os.path.relpath(folder, root_dir)
                rel_file = os.path.join(rel_dir, file) if rel_dir != '.' else file
                code_files.append((rel_dir, file, os.path.join(folder, file)))
    return code_files

def main():
    root_dir = os.path.dirname(os.path.abspath(__file__))
    code_files = collect_code_files(root_dir)
```

All Code Files

```
    code_files.sort()  # Sort for consistent order

    pdf = PDF()
    current_folder = None
    for rel_dir, file, path in code_files:
        if rel_dir != current_folder:
            pdf.chapter_title(f"Folder: {rel_dir}")
            current_folder = rel_dir
        pdf.chapter_title(f"File: {file}")
        try:
            with open(path, 'r', encoding='utf-8') as f:
                code = f.read()
        except Exception as e:
            code = f"[Could not read file: {e}]"
        pdf.code_block(code)

    pdf.output(OUTPUT_PDF)
    print(f"PDF generated: {OUTPUT_PDF}")

if __name__ == "__main__":
    main()
```

Folder: 01. Product Class

File: Cars.java

```
package MODULE1;

import java.util.Scanner;


public class Cars {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter the number of products: ");
        int numProducts = sc.nextInt();
        sc.nextLine();


        Product[] products = new Product[numProducts];


        for (int i = 0; i < numProducts; i++) {
            System.out.println("\nEnter details for product " + (i + 1) + ":");

            System.out.print("Enter product code: ");
            String code = sc.nextLine();

            System.out.print("Enter product name: ");
```

All Code Files

```
            String name = sc.nextLine();

            System.out.print("Enter product price: ");
            int price = sc.nextInt();
            sc.nextLine();


            products[i] = new Product(name, code, price);
        }


        System.out.println("\nDisplaying details of all products:");
        for (Product p : products) {
            p.display();
        }


        Product lowestPriceProduct = products[0];
        for (int i = 1; i < numProducts; i++) {
            if (products[i].getprice() < lowestPriceProduct.getprice()) {
                lowestPriceProduct = products[i];
            }
        }


        System.out.println("\nDisplaying product with the lowest price:");
        lowestPriceProduct.display();

        sc.close();
    }
}
```

File: Product.java

```
package MODULE1;


public class Product {
 String pname,pcode;
 int price;
 public Product() {}
 public Product(String pname,String pcode,int price) {

  this.pname = pname;
  this.pcode = pcode;
  this.price = price;}
 public void setpname(String pname) {
  this.pname = pname;}
 public String getpcode() {
  return pcode;}
 public String getpname() {
```

All Code Files

```
  return pname;}
 public void setpcode(String pcode) {
  this.pcode=pcode ;}
 public int getprice() {
  return price ;}
 public void setprice(int price) {
  this.price=price ;}
 public void display( ) {
  System.out.println("pcode :" + this.pcode);
  System.out.println("pname:" + this.pname);
  System.out.println("price :" + this.price + "\n");


 }

}
```

Folder: 02. Matrix Addition

File: MatrixAddition.java

```
package MODULE1;

import java.util.Scanner;

public class MatrixAddition {
 @SuppressWarnings("resource")
 public static void main(String[] args) {
  int p,q,m,n;
  Scanner s = new Scanner(System.in);
  System.out.print("Enter rows of Matrix A ");
  p = s.nextInt();
  System.out.print("Enter columns of Matrix A ");
  q = s.nextInt();
  System.out.print("Enter rows of Matrix B ");
  m = s.nextInt();
  System.out.print("Enter columns of Matrix B ");
  n = s.nextInt();
  if(p == m && q == n)
  {
   int a[][] = new int[p][q];
   int b[][] = new int[m][n];
   System.out.println("Enter elements of Matrix A");
   for(int i = 0; i < p; i++)
    for(int j = 0; j < q; j++)
     a[i][j] = s.nextInt();
   System.out.println("Enter elements of Matrix B");
   for(int i = 0; i < m; i++)
    for(int j = 0; j < n; j++)
     b[i][j] = s.nextInt();
```

All Code Files

```
   System.out.println("Matrix A");
   for(int i = 0; i < p; i++) {
    for(int j = 0; j < q; j++)
     System.out.print(a[i][j] + "  ");
    System.out.println();
   }
   System.out.println("Matrix B");
   for(int i = 0; i < m; i++) {
    for(int j = 0; j < n; j++)
     System.out.print(b[i][j] + "  ");
    System.out.println();
   }

   System.out.println("Sum Matrix");
   for(int i = 0; i < m; i++)
   {
    for(int j = 0; j < n; j++)
     System.out.print((a[i][j] + b[i][j]) + "  ");
    System.out.println();
   }
  }
  else {
   System.out.println("These Matrices cannot be added");
  }
 }
}
```

Folder: 03. Addition Complex Number

File: ComplexNumber.java

```
package MODULE1;

import java.util.Scanner;

public class ComplexNumber {
 double real ,img ;
 ComplexNumber(double r, double i)
 {
  real=r; img =i;
 }
 public static ComplexNumber sum(ComplexNumber c1,ComplexNumber c2)
 {
  ComplexNumber temp = new ComplexNumber(0,0) ;
  temp.real =c1.real + c2.real;
  temp.img = c1.img+c2.img;
  return temp;
 }
 public static void main(String[] args)
 {
  double p,q,m,n;
```

All Code Files

```
  Scanner s = new Scanner(System.in);
  System.out.print ("Enter the first real part :");
  p=s.nextInt();
  System.out.print ("Enter the first img part :");
  q=s.nextInt();
  System.out.print ("Enter the second real part :");
  m=s.nextInt();
  System.out.print ("Enter the second img part :");
  n=s.nextInt();
  ComplexNumber c1 = new ComplexNumber(p,q);
  ComplexNumber c2 = new ComplexNumber(m,n);
  ComplexNumber temp = sum(c1,c2);
  System.out.printf("sum is: "+ temp.real+ "+" +temp.img + "i");
 }

}
```

Folder: 04. Matrix Symmetry

File: MatrixSymmetry.java

```
package MODULE1;

import java.util.Scanner;

public class MatrixSymmetry {

 @SuppressWarnings("resource")
 public static void main(String[] args) {
  int rows,cols;
  Scanner s = new Scanner(System.in);
  System.out.print("Enter rows of Matrix A ");
  rows = s.nextInt();
  System.out.print("Enter columns of Matrix A ");
  cols = s.nextInt();
  int matrix[][] = new int[rows][cols];
  System.out.println("Enter elements");
  for(int i = 0 ; i < rows; i++)
  {
   for(int j = 0; j < cols; j++)
    matrix[i][j] = s.nextInt();
  }
  System.out.println("Given Matrix");
  for(int i = 0 ; i < rows; i++)
  {
   for(int j = 0; j < cols; j++)
    System.out.print(matrix[i][j] + "\t");
   System.out.println();
  }
```

All Code Files

```
  if(rows != cols)
   System.out.println("Given Matrix is not source matrix");
  else {
   boolean sym = true;
   for(int i = 0; i < rows; i++)
    for(int j = 0; j < cols; j++)
     if(matrix[i][j] != matrix[j][i]) {
      sym = false;
      break;
     }
   if(sym)
    System.out.println("Matrix is Symmetric");
   else
    System.out.println("Matrix is not Symmetric");
  }



 }

}
```

Folder: 05. CPU Details

File: CPU.java

```
package MODULE1;
public class CPU{
 double price;
 class Processor{
  double cores;
  String Manufacturer;
  double Cache;
  public Processor(double cache)
  {
   Cache = cache;
  }
  double getCache() {
   return Cache;
  }
 }
 static class Ram{
  double memory,clockSpeed;
  String Manufacurer;
  Ram(double cs)
  {
   clockSpeed = cs;
  }
  double getClockSpeed() {
   return clockSpeed;
  }
```

All Code Files

```
 }

}
```

File: CPUDetails.java

```
package MODULE1;

import java.util.Scanner;

public class CPUDetails{
 public static void main(String[] args) {
  CPU cpu = new CPU();
  Scanner s = new Scanner(System.in);
  System.out.println("What's cache of your processor");
  double cache = s.nextDouble();
  System.out.println("Ram ClockSpeed?");
  double clockSpeed = s.nextDouble();
  CPU.Processor p = cpu.new Processor(cache);
  CPU.Ram ram = new CPU.Ram(clockSpeed);
  s.close();
  System.out.println("Processor Cache  "+p.getCache());
  System.out.println("Ram Clock Speed  "+ram.getClockSpeed()+" GHz");
 }

}
```

Folder: 06. Sort Strings

File: StringSort.java

```
package MODULE2;

import java.util.Arrays;
import java.util.Scanner;

public class StringSort {


  public static void main(String[] args) {
   int count = 0;
   String tmp;
   Scanner scan =new Scanner(System.in);
   System.out.println("Enter the number of string to sort");
   count= scan.nextInt();
   String strlist[]=new String [count];
   Scanner scan1=new Scanner(System.in);
   System.out.println("Enter your string");
   for(int i=0;i<count;i++)
    strlist[i] =scan1.nextLine();
   System.out.println("choose 1 or 2 from the menu below");
   System.out.println("1: in-built sort");
```

All Code Files

```
   System.out.println("2:user-defined sort");
   int choice;
   choice=scan.nextInt();
   switch (choice) {
   case 1 :Arrays.sort(strlist);
   System.out.println(Arrays.toString(strlist));
   break;
   case 2: for(int i=0;i<count-1;i++)
    for(int j=i+1;i<strlist.length;j++)
     if(strlist[i].compareTo(strlist[j])>0){
      tmp=strlist[i];
      strlist[i]=strlist[j];
      strlist[j]=tmp;}
   System.out.println(Arrays.toString(strlist));
   break;









   }



 }

}
```

Folder: 07. Linear Search

File: LinearSearch.java

```
package MODULE2;

import java.util.Scanner;

public class LinearSearch {

 public static void main(String[] args) {
  int c,n,search,array[];
  Scanner in =new Scanner(System.in);
  System.out.println("Enter the number of elements");
  n= in.nextInt();
  array =new int[n];
  System.out.println("Enter those "+ n +" elements");
  for(c=0;c<n;c++)
   array[c]= in.nextInt();
  System.out.println("Enter the value you want to find:");
  search= in.nextInt();
  for(c=0;c<n;c++)
   if(array[c]==search) {
```

All Code Files

```
    System.out.println(search +"is present in the location:" +(c+1));
    break;
   }
  if(c==n)
   System.out.println(search +"is not present in the array");


 }

}
```

Folder: 08. String Manipulation

File: StringManipulation.java

```
package MODULE2;

public class StringManipulation {

 public static void main(String[] args) {
  char arrSample[] = {'R','O','S','E'};
  String strSample1 = new String (arrSample);
  System.out.println(strSample1+"\t");
  System.out.println("Length of "+strSample1+" is "+strSample1.length());
  System.out.println();
  byte ascii[] = {65,66,67,68,69,70};
  String strSample2 = new String (ascii);
  System.out.println(strSample2+"\t");
  String strSampel3 = strSample1.concat(strSample2);
  System.out.println(strSampel3+"\t");
  String strSample4 = strSample1 + strSample2;
  System.out.println(strSample4+"\t");
  System.out.println("3rd char of "+strSample4+" is "+strSample4.charAt(2));
  char byt[] = new char[3];
  strSample4.getChars(2, 5, byt, 0);
  System.out.println("Extracted chars "+strSample4+" "+new String(byt));
  String strSample = "RockStar";
  System.out.println(strSample.compareTo("ROCKSTAR")+"\t");
  System.out.println(strSample.compareTo("rockstar")+"\t");
  System.out.println(strSample.equals("ROCKSTAR"));
  System.out.println(strSample.equalsIgnoreCase("ROCKSTAR"));
  System.out.println("RockStar starts with 'RO "+strSample.startsWith("RO"));
  System.out.println("RockStar contains 'ta'"+strSample.contains("ta"));
  System.out.println("RockStar ends with 'star'"+strSample.endsWith("star"));
  System.out.println("Index of 't' in RockStar: "+strSample.indexOf("t"));
  System.out.println("All caps-RockStar: "+strSample.toUpperCase());
  System.out.println("All lowcase-RockStar: "+strSample.toLowerCase());
  System.out.println("Replace Star with et: "+strSample.replace("Star", "et"));
  String dVal = "3.456";
  String iVal = "6";
  System.out.println("Double value of dVal is: "+Double.valueOf(dVal));
```

All Code Files

```
  System.out.println("Integer value of iVal is: "+Integer.valueOf(iVal));
  double a = Double.valueOf(dVal);
  int b = Integer.valueOf(iVal);
  System.out.println(a+b);
 }

}
```

Folder: 09. Employee Info And Search

File: Employee.java

```
package MODULE2;
import java.util.Scanner;

public  class Employee {
 int eNo, eSalary;
 String eName;
 public Employee(int no, int sal, String name) {
  this.eNo = no;
  this.eSalary = sal;
  this.eName = name;
 }
 public void showData(){
  System.out.println("EmpId: "+eNo+"\nName: "+eName+"\nSalary: "+eSalary);
  System.out.println();
 }
```

```
 public static class EmpArrObjects {

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

   System.out.println("Enter number of employees");
   Scanner sc = new Scanner(System.in);
   int n = sc.nextInt();
   System.out.println("Enter employee details: ");
   Employee employees[] = new Employee[n];
   Scanner scemp = new Scanner(System.in);
   int eid,esal;
   String  ename;
   for(int i = 0; i  < n; i++) {
    System.out.println("\nEnter details of employee "+(i+1));
    System.out.println("Employee id(integer): ");
```

```
    eid = scemp.nextInt();
    scemp.nextLine();
    System.out.println("Employee name: ");
    ename = scemp.nextLine();
    System.out.println("Salary: ");
    esal = scemp.nextInt();
    Employee emp = new Employee(eid, esal, ename);
    employees[i] = emp;
   }
   System.out.println("Employees are:\n");
   for(Employee y: employees)
    y.showData();
   System.out.println("Enter employee number to search");
   int search = sc.nextInt();
   boolean found = false;
   for(Employee x: employees) {
    if(search == x.eNo) {
     found = true;
     System.out.println("Employee found");
     x.showData();
     break;
    }
   }
   if(!found)
    System.out.println("Employee not found");
   sc.close();
   scemp.close();
  }

 }
}
```

Folder: 10. Method Overloading Area

File: Overload.java

```
package MODULE2;

import java.util.Scanner;

public class Overload {
 public static void main(String[] args) {

  OverloadDemo obj = new OverloadDemo();
  Scanner scanner = new Scanner(System.in);
  System.out.println("Enter Size of Square: ");
  int size = scanner.nextInt();
  System.out.println("Enter Length of Rectangle");
  int l = scanner.nextInt();
  System.out.println("Enter width of Rectangle");
  int w = scanner.nextInt();
```

All Code Files

```
  System.out.println("Enter Side of Triangle");
  double side = scanner.nextDouble();
  obj.area(size);
  obj.area(w, l);
  obj.area(side);
 }

}
```

File: OverloadDemo.java

```
package MODULE2;

public class OverloadDemo {
 void area(long x) {
  long a = x * x;
  System.out.println("Area of Square is: "+a+" sq units");
 }
 void area(float x, float y) {
  float a = x * y;
  System.out.println("Area of Rectangle is: "+a+" sq units");
 }
 void area(double x) {
  double a = Math.PI*x*x;
  System.out.println("Area of Circle is: "+a+" sq units");
 }
}
```

Folder: 11. Teacher Class Inheritance

File: EmployeeT.java

```
package MODULE3;

public class EmployeeT {
 int empid;
 String name,address;
 float salary;
 public EmployeeT(int empid, String name, String address, float salary) {
  this.empid = empid;
  this.name = name;
  this.address = address;
  this.salary = salary;
 }
}
```

File: Teacher.java

```
package MODULE3;

public class Teacher extends EmployeeT{
 String department,subject;
```

All Code Files

```
 Teacher(int empid, String name, String address, float salary, String department, String
subject){
  super(empid,name,address,salary);
  this.department = department;
  this.subject = subject;
 }
 protected void display(){
  System.out.println("Teacher id: "+empid);
  System.out.println("Teacher name: "+name);
  System.out.println("Teacher salary: "+salary);
  System.out.println("Teacher Address: "+address);
  System.out.println("Teacher department: "+department);
  System.out.println("Teacher subject: "+subject);
 }
}
```

File: TeacherArrObjects.java

```
package MODULE3;

import java.util.Scanner;

public class TeacherArrObjects {
 public static void main(String[] args) {
  System.out.println("Enter number of teachers: ");
  Scanner sc = new Scanner(System.in);
  int n = sc.nextInt();
  Teacher teacher[] = new Teacher[n];
  Scanner scT = new Scanner(System.in);
  int tid;
  float salary;
  String name,address,department,subject;
  for(int i = 0; i < n; i++) {
   System.out.println("Enter teacher-"+(i+1)+"'s details");
   System.out.println("Enter Teacher id (integer): ");
   tid = scT.nextInt();
   scT.nextLine();
   System.out.println("Enter Name: ");
   name = scT.nextLine();
   System.out.println("Enter address: ");
   address = scT.nextLine();
   System.out.println("Enter salary (float): ");
   salary = scT.nextFloat();
   scT.nextLine();
   System.out.println("Enter department: ");
   department = scT.nextLine();
   System.out.println("Enter subject: ");
   subject = scT.nextLine();
   Teacher t = new Teacher(tid, name, address, salary, department, subject);
   teacher[i] = t;
  }
```

All Code Files

```
  System.out.println("\n-----------------------------------\n");
  System.out.println("Teachers are: ");
  for(Teacher x: teacher) {
   x.display();
   System.out.println("\n");
  }
  System.out.println("-----------------------------------\n");
  sc.close();
  scT.close();
 }

}
```

Folder: 12.AreaAndPerimeterOfObjects

File: AP.java

```
package MODULE3;

public interface AP
{
 void input();
 void area();
 void perimeter();
}
```

File: Circle.java

```
package MODULE3;

import java.util.Scanner;

public class Circle implements AP
{
 int r = 0;
 double pi = 3.14, area = 0, perimeter = 0;

 public void input()
 {
  Scanner c = new Scanner(System.in);
  r = c.nextInt();
 }

 public void area()
 {
  area = pi*r*r;
  System.out.println("Area of the circle:"+area);
 }
 public void perimeter()
 {
  perimeter = 2*pi*r;
  System.out.println("Perimeter of circle:"+perimeter);
```

All Code Files

```
 }
}
```

File: IFaceDemo.java

```
package MODULE3;

import java.util.Scanner;

public class IFaceDemo
{

 public static void main(String[] args)
 {
  int ch;
  Circle c = new Circle();
  Rectangle r = new Rectangle();
  Scanner sc = new Scanner(System.in);

  System.out.println("Options\n1 : Area of the circle");
  System.out.println("2 : Area of the rectangle");
  System.out.println("3 : Perimeter of the circle");
  System.out.println("4 : Perimeter of the rectangle");
  System.out.println("5 : Program termination");

  lp : while(true)
  {
   System.out.println("Enter your option:");
   ch = sc.nextInt();

   switch(ch)
   {
   case 1:
    System.out.print("Enter the radius of the circle:");
    c.input();
    c.area();
    break;

   case 2:
    System.out.print("Enter the length and breadth of the rectangle:");
    r.input();
    r.area();
    break;

   case 3:
    System.out.print("Enter the radius of the circle:");
    c.input();
    c.perimeter();
    break;

   case 4:
```

All Code Files

```
    System.out.print("Enter the length and breadth of the rectangle:");
    r.input();
    r.perimeter();
    break;

   case 5:
    System.out.print("Terminating the program....");
    break lp;

   default:
    System.out.println("Invalid option!! Select a valid option..");
   }
  }

  sc.close();

 }

}
```

File: Rectangle.java

```
package MODULE3;

import java.util.Scanner;

public class Rectangle implements AP
{
 int l = 0,b = 0;
 double area,perimeter;

 public void input()
 {
  Scanner c = new Scanner(System.in);
  l = c.nextInt();
  b = c.nextInt();
 }

 public void area()
 {
  area = l*b;
  System.out.println("Area of the rectangle:"+area);
 }
 public void perimeter()
 {
  perimeter = 2*(l+b);
  System.out.println("Perimeter of rectangle:"+perimeter);
 }
}
```

Folder: 13.Multilevel Inheritance

All Code Files

File: Vehicle.java

```
package MODULE3;

import java.util.Scanner;

class Vehicle {
    String registrationNumber;
    int speed;
    int fuelCapacity;

    Vehicle() {}

    Vehicle(String registrationNumber, int speed, int fuelCapacity) {
        this.registrationNumber = registrationNumber;
        this.speed = speed;
        this.fuelCapacity = fuelCapacity;
    }

    public void displayInfo() {
        System.out.println("Registration Number: " + registrationNumber);
        System.out.println("Speed: " + speed + " km/h");
        System.out.println("Fuel Capacity: " + fuelCapacity + " liters");
    }
}

class Car extends Vehicle {
    int numberOfDoors;
    boolean hasAirConditioning;

    Car() {}

       Car(String registrationNumber, int speed, int fuelCapacity, int numberOfDoors,
boolean hasAirConditioning) {
        super(registrationNumber, speed, fuelCapacity);
        this.numberOfDoors = numberOfDoors;
        this.hasAirConditioning = hasAirConditioning;
    }

    public void displayCarInfo() {
        displayInfo();
        System.out.println("Number of Doors: " + numberOfDoors);
        System.out.println("Air Conditioning: " + (hasAirConditioning ? "Yes" : "No"));
    }
}

class ElectricCar extends Car {
    int batteryCapacity; // in kWh
    int chargeLevel;     // in percentage

    ElectricCar(String registrationNumber, int speed, int fuelCapacity,
```

```
                int numberOfDoors, boolean hasAirConditioning,
                int batteryCapacity, int chargeLevel) {
                      super(registrationNumber,   speed,   fuelCapacity,   numberOfDoors,
hasAirConditioning);
        this.batteryCapacity = batteryCapacity;
        this.chargeLevel = chargeLevel;
    }

    public void displayElectricCarInfo() {
        displayCarInfo();
        System.out.println("Battery Capacity: " + batteryCapacity + " kWh");
        System.out.println("Charge Level: " + chargeLevel + "%");
    }
}
```

File: VehicleArr.java

```
package MODULE3;
import java.util.Scanner;

public class VehicleArr {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of electric cars: ");
        int n = sc.nextInt();

        ElectricCar[] cars = new ElectricCar[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Electric Car " + (i + 1) + ":");

            System.out.print("Registration Number: ");
            String regNum = sc.next();

            System.out.print("Speed (km/h): ");
            int speed = sc.nextInt();

            System.out.print("Fuel Capacity (liters): ");
            int fuelCap = sc.nextInt();

            System.out.print("Number of Doors: ");
            int doors = sc.nextInt();

            System.out.print("Has Air Conditioning? (true/false): ");
            boolean ac = sc.nextBoolean();

            System.out.print("Battery Capacity (kWh): ");
            int batteryCap = sc.nextInt();
```

All Code Files

```
            System.out.print("Charge Level (%): ");
            int charge = sc.nextInt();

               cars[i] = new ElectricCar(regNum, speed, fuelCap, doors, ac, batteryCap,
charge);
        }

        System.out.println("\nElectric Cars Details:\n");
        for (ElectricCar car : cars) {
            car.displayElectricCarInfo();
        }

    }
}
```

Folder: 14.BillGeneration

File: Bill.java

```
package MODULE3;

import java.util.*;

interface BillGen
{
    int calculate();
}

class ProductB implements BillGen
{
    String name;
    int prod_id, quantity, unit_price, total;

    ProductB(String n, int p, int q, int u) {
        name = n;
        prod_id = p;
        quantity = q;
        unit_price = u;
    }

    public int calculate() {
        total = quantity * unit_price;
        return total;
    }
}

public class Bill
{
    public static void main(String[] args)
    {
```

All Code Files

```
        ProductB[][] order;
        System.out.println("Enter no. of orders:");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        order = new ProductB[n][];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter no. of products:");
            int m = sc.nextInt();
            order[i] = new ProductB[m];

            Set<Integer> productIds = new HashSet<>();

            for(int j = 0; j < m; j++)
            {
                System.out.println("Enter Product " + (j + 1) + " name:");
                String a = sc.next();

                int b;
                while (true)
                {
                    System.out.println("Enter Product ID:");
                    b = sc.nextInt();

                    if (productIds.contains(b))
                    {
                         System.out.println("Error: Product ID " + b + " is already used!
Please enter a unique ID.");
                    }
                    else
                    {
                        productIds.add(b);
                        break;
                    }
                }

                System.out.println("Enter Product quantity:");
                int c = sc.nextInt();
                System.out.println("Enter Product unit price:");
                int d = sc.nextInt();

                ProductB pb = new ProductB(a, b, c, d);
                order[i][j] = pb;
                order[i][j].total = order[i][j].calculate();
            }

        }

        for (int i = 0; i < n; i++) {
            int sum = 0;
            System.out.println("Order no: " + (i + 1));
```

All Code Files

```
            Date date = java.util.Calendar.getInstance().getTime();
            System.out.println(date);

System.out.println("-------------------------------------------------------------------------
-------------------------------------------------");
             System.out.printf("%5s %20s %25s %10s %10s", "Prod_Id", "Name", "Quantity",
"Unit_price", "Total");
            System.out.println();

System.out.println("-------------------------------------------------------------------------
-------------------------------------------------");

            for (int j = 0; j < order[i].length; j++) {
                      System.out.printf("%5s %20s %25s %10s %10s", order[i][j].prod_id,
order[i][j].name, order[i][j].quantity, order[i][j].unit_price, order[i][j].total);
                System.out.println();
            }


System.out.println("-------------------------------------------------------------------------
-------------------------------------------------");
            for (int k = 0; k < order[i].length; k++) {
                sum = sum + order[i][k].total;
            }
            System.out.println("Net Amount: " + sum);

System.out.println("-------------------------------------------------------------------------
-------------------------------------------------");
        }
        sc.close();
    }
}
```

Folder: 15.InheritanceBook

File: InheritanceBook.java

```
package MODULE_3;

class Publisher
{
 String publisher;
 Publisher(String publi)
 {
  this.publisher = publi;
 }
}

class Book
{
 String name;
```

All Code Files

```
 Publisher publisher;
 public Book(String name,Publisher publisher)
 {
  this.name = name;
  this.publisher = publisher;
 }
}

class Literature extends Book
{
 String Lit_type = "Literature";
 Literature(String name,Publisher publisher)
 {
  super(name,publisher);
 }
 void display()
 {
  System.out.println("Name : "+super.name);
  System.out.println("Type : "+this.Lit_type);
  System.out.println("Publisher : "+this.publisher.publisher);
 }
}


class Fiction extends Book
{
 String Lit_type = "Fiction";
 Fiction(String name,Publisher publisher)
 {
  super(name,publisher);
 }
 void display()
 {
  System.out.println("Name : "+super.name);
  System.out.println("Type : "+this.Lit_type);
  System.out.println("Publisher : "+this.publisher.publisher);
 }
}


public class InheritanceBook
{

 public static void main(String[] args)
 {
  Publisher lp = new Publisher("S.Chand");
  Literature l = new Literature("As you like it",lp);
  l.display();
  System.out.println("-----------------");
  Publisher fp = new Publisher("Tata McGraw Hill");
  Fiction f = new Fiction("Tempest",fp);
```

All Code Files

```
  f.display();
 }

}
```

Folder: 16.ImplementInterface

File: SportStudentResult.java

```
package MODULE_3;
interface Student
{
 int score = 0;
 void displayScore();
}

interface Sports
{
 int score = 25;
 void displaySportsScore();
}

class Result implements Student,Sports
{
 public void displayScore()
 {
  System.out.println("Academic score is:"+Student.score);
 }
 public void displaySportsScore()
 {
  System.out.println("Sports Score is:"+Sports.score);
 }
}

public class SportStudentResult {

 public static void main(String[] args) {

  Result r = new Result();
  r.displayScore();
  r.displaySportsScore();
 }
}
```

Folder: 17.Packages\figures

File: Circle.java

```
package figures;

public class Circle
{
```

All Code Files

```
 private double radius;
 public Circle(double r) {radius = r;}
 public double perimeter() {return (2*3.14*radius);}
 public double area() {return (3.14*radius*radius);}
}
```

File: FigureCi.java

```
package figures;

public class FigureCi
{
 double perimeter() {
  return 0;
 }
 double area() {
  return 0;
 }
}
```

File: FigureSq.java

```
package figures;
public interface FigureSq
{
 int perimeter();
 int area();
}
```

File: FigureTr.java

```
package figures;

public interface FigureTr
{
 int perimeter();
 double area();
}
```

File: Square.java

```
package figures;

public class Square implements FigureSq
{
 private int side;
 public Square(int s) {side = s;}
 public int perimeter() {return (4*side);}
 public int area() {return (side*side);}
}
```

All Code Files

File: Triangle.java

```
package figures;

public class Triangle implements FigureTr
{
 private int side1,side2,side3;
 public Triangle(int s1,int s2,int s3)
 {
  side1 = s1;
  side2 = s2;
  side3 = s3;
 }
 public int perimeter()
 {
  return side1+side2+side3;
 }
 public double area()
 {
  double s = (side1+side2+side3)/2;
  double a = Math.sqrt((s-side1)+(s-side2)+(s-side3));
  return a;
 }
}
```

Folder: 18.ArithmeticPackage\Arithmetic

File: Add.java

```
package Arithmetic;

public class Add implements OprIF
{
 private int num1,num2;

 public Add(int num1,int num2)
 {
  this.num1 = num1;
  this.num2 = num2;
 }
 public int opr()
 {
  return num1+num2;
 }



}
```

File: Div.java

```
package Arithmetic;
```

All Code Files

```
public class Div implements DivIF
{
 private int num1,num2;
 public Div(int num1,int num2)
 {
  this.num1 = num1;
  this.num2 = num2;
 }
 public float opr()
 {
  return num1/num2;
 }

}
```

File: DivIF.java

```
package Arithmetic;

public interface DivIF
{
 float opr();
}
```

File: Mul.java

```
package Arithmetic;

public class Mul implements OprIF
{
 private int num1,num2;

 public Mul(int num1,int num2)
 {
  this.num1 = num1;
  this.num2 = num2;
 }
 public int opr()
 {
  return num1*num2;
 }


}
```

File: OprIF.java

```
package Arithmetic;

public interface OprIF
{
 int opr();
}
```

All Code Files

File: Sub.java

```
package Arithmetic;

public class Sub implements OprIF
{
 private int num1,num2;

 public Sub(int num1,int num2)
 {
  this.num1 = num1;
  this.num2 = num2;
 }
 public int opr()
 {
  return num1-num2;
 }


}
```

Folder: 18.ArithmeticPackage\Test

File: Calc.java

```
package Test;
import java.util.Scanner;
import Arithmetic.*;
public class Calc
{

 public static void main(String[] args)
 {
  System.out.println("Calculator");
  System.out.println("-----------");
  int num1,num2;
  Scanner s = new Scanner(System.in);

  int option=0;
  System.out.println("Operations:\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Divisi
on\n5.Exit");
  while(option!=5)
  {
   System.out.println("Enter your option:");
   option = s.nextInt();

   switch(option)
   {
   case 1:
```

All Code Files

```
    System.out.println("Enter number 1:");
    num1 = s.nextInt();
    System.out.println("Enter number 2:");
    num2 = s.nextInt();
    Add add = new Add(num1,num2);
    System.out.println("Sum of "+num1+" & "+num2+" is:"+add.opr());
    break;
   case 2:
    System.out.println("Enter number 1:");
    num1 = s.nextInt();
    System.out.println("Enter number 2:");
    num2 = s.nextInt();
    Sub sub = new Sub(num1,num2);
    System.out.println("Difference of "+num1+" & "+num2+" is:"+sub.opr());
    break;

   case 3:
    System.out.println("Enter number 1:");
    num1 = s.nextInt();
    System.out.println("Enter number 2:");
    num2 = s.nextInt();
    Mul mul = new Mul(num1,num2);
    System.out.println("Product of "+num1+" & "+num2+" is:"+mul.opr());
    break;

   case 4:
    System.out.println("Enter number 1:");
    num1 = s.nextInt();
    System.out.println("Enter number 2:");
    num2 = s.nextInt();
    Div div = new Div(num1,num2);
    System.out.println("Division of "+num1+" & "+num2+" is:"+div.opr());
    break;

   case 5:
    System.out.println("Exiting....");
    break;

   default:
    System.out.println("Enter a valid option!!!!!!!!");
   }
  }
  s.close();
 }

}
```