Week 5

1.Create a “circle” class & a “point” class. The coordinates of the circle are given and used within the “circle” class as object of the “point” class. Display the area of circle.

Code:

```
import java.lang.Math;


class Point {
    private double x;
    private double y;

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }
}

class Circle {
    private Point center;
    private double radius;

    public Circle(Point center, double radius) {
        this.center = center;
        this.radius = radius;
    }

    public double area() {
        return Math.PI * radius * radius;
    }
}

public class HelloWorld {
    public static void main(String[] args) {
        Point centerPoint = new Point(0, 0);
        double radius = 5;
        Circle circle = new Circle(centerPoint, radius);
        System.out.println("Area of the circle: " + circle.area());
    } }
```

```
Area of the circle: 78.53981633974483
PS D:\JAVA_Lab_Assignments>
```

2.Create a class called Time, which has three private instance variables – hour, min and sec. It contains a method called add( ) which takes one Time object as parameter and print the added value of the calling Time object and passes Time object. In the main method, declare two Time objects and assign values using constructor and call the add() method

Code:

```
public class Time {
    private int hour;
    private int minute;
    private int second;
    public Time(int hour, int minute, int second) {
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }
    public void add(Time otherTime) {
        int totalSeconds = this.hour * 3600 + this.minute * 60 + this.second;
        totalSeconds += otherTime.hour * 3600 + otherTime.minute * 60 +
otherTime.second;

        int hours = totalSeconds / 3600;
        totalSeconds %= 3600;
        int minutes = totalSeconds / 60;
        int seconds = totalSeconds % 60;

        System.out.println("Added Time: " + hours + " hours " + minutes + " minutes " +
seconds + " seconds");
    }
    public static void main(String[] args) {
        Time time1 = new Time(5, 30, 45);
        Time time2 = new Time(2, 15, 10);
        time1.add(time2);
    }
}
```

```
Added Time: 7 hours 45 minutes 55 seconds
PS D:\JAVA_Lab_Assignments>
```

3. Create a class called Complex, which has three private instance variables – real and imaginary. It contains a method called add( ) which takes one Complex object as parameter and print the added value of the calling Complex object and passes Complex object. In the main method, declare two Complex objects and assign values using constructor and call the add() method.

Code:

```
public class Complex {
    private double real;
    private double imaginary;
    public Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }
    public void add(Complex otherComplex) {
```

```
        double sumReal = this.real + otherComplex.real;
        double sumImaginary = this.imaginary + otherComplex.imaginary;
        System.out.println("Sum: " + sumReal + " + " + sumImaginary + "i");
    }
    public static void main(String[] args) {
        Complex complex1 = new Complex(3.5, 2.0);
        Complex complex2 = new Complex(1.5, 4.0);
        complex1.add(complex2); }
}
```

```
Sum: 5.0 + 6.0i
```

4.Write a program to define a class having one 3-digit number, num as data member. Initialize and display reverse of that number.

```
import java.util.Scanner;
class Number {
   private int num;
   public Number(int num) {
      if (num >= 100 && num <= 999) {
         this.num = num;
      } else {
         throw new IllegalArgumentException("Number must be a 3-digit number");       } }
   public int reverseNumber() {
      int reversed = 0;
      int temp = num;
      while (temp != 0) {
         int digit = temp % 10;
         reversed = reversed * 10 + digit;
         temp /= 10;       }
      return reversed;    }
   public void displayReverse() {
      System.out.println("Original Number: " + num);
      System.out.println("Reverse of the Number: " + reverseNumber());    }}
public class Num {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      try {
         System.out.print("Enter a 3-digit number: ");
         int num = scanner.nextInt();
         Number number = new Number(num);
         number.displayReverse();
      } catch (IllegalArgumentException e) {
         System.out.println(e.getMessage());
      } finally {
         scanner.close();      }   }}
```

```
Enter a 3-digit number: 875
Original Number: 875
Reverse of the Number: 578
PS C:\Users\burni\OneDrive\Desktop\java>
```

5. Write a program to define a class Student with four data members such as name, roll no., sub1, and sub2. Define appropriate methods to initialize and

display the values of data members. Also calculate total marks and percentage scored by student.

Code:

```
public class Student {
    private String name;
    private int rollNo;
    private int sub1;
    private int sub2;
    public Student(String name, int rollNo, int sub1, int sub2) {
        this.name = name;
        this.rollNo = rollNo;
        this.sub1 = sub1;
        this.sub2 = sub2;
    }
    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Roll No.: " + rollNo);
        System.out.println("Marks in Subject 1: " + sub1);
        System.out.println("Marks in Subject 2: " + sub2);
        System.out.println("Total Marks: " + calculateTotalMarks());
        System.out.println("Percentage: " + calculatePercentage() + "%");
    }
    private int calculateTotalMarks() {
        return sub1 + sub2;
    }
    private double calculatePercentage() {
        int totalMarks = calculateTotalMarks();
        return (totalMarks / 2.0);
    }
    public static void main(String[] args) {
        Student student1 = new Student("John", 101, 85, 90);
        student1.displayDetails();
    }
}
```

```
Name: John
Roll No.: 101
Marks in Subject 1: 85
Marks in Subject 2: 90
Total Marks: 175
Percentage: 87.5%
```

6. Write a program to define a class Employee to accept emp_id, emp _name, basic_salary from the user and display the gross_salary.

```
import java.util.Scanner;
class Employee {
   int emp_id;
   String emp_name;
   double basic_salary;
   Employee(int id, String name, double salary) {
      emp_id = id;
```

```
        emp_name = name;
        basic_salary = salary;    }
    double calculateGrossSalary() {
        double hra = 0.2 * basic_salary;
        double da = 0.1 * basic_salary;
        double gross_salary = basic_salary + hra + da;
        return gross_salary;    }
    void displayDetails() {
        double gross_salary = calculateGrossSalary();
        System.out.println("Employee ID: " + emp_id);
        System.out.println("Employee Name: " + emp_name);
        System.out.println("Basic Salary: " + basic_salary);
        System.out.println("Gross Salary: " + gross_salary);    }}
public class salary {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Employee ID: ");
        int emp_id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Employee Name: ");
        String emp_name = scanner.nextLine();
        System.out.print("Enter Basic Salary: ");
        double basic_salary = scanner.nextDouble();
        Employee emp = new Employee(emp_id, emp_name, basic_salary);
        emp.displayDetails();
        scanner.close();   }}
```

```
Enter Employee ID: 1234
Enter Employee Name: burnik
Enter Basic Salary: 5000
Employee ID: 1234
Employee Name: burnik
Basic Salary: 5000.0
Gross Salary: 6500.0
PS C:\Users\burni\OneDrive\Desktop\java>
```

7. Write a program to define a class Fraction having data members numerator and denominator. Initialize three objects using different constructors and display its fractional value.

Code:

```
public class Fraction {
    private int numerator;
    private int denominator;
    public Fraction(int numerator, int denominator) {
        this.numerator = numerator;
        this.denominator = denominator;
    }
    public Fraction(int numerator) {
        this.numerator = numerator;
        this.denominator = 1;
    }
    public Fraction() {
```

```
        this.numerator = 1;
        this.denominator = 1;
    }
    public void displayFraction() {
        System.out.println(numerator + "/" + denominator);
    }
    public static void main(String[] args) {
        Fraction fraction1 = new Fraction(3, 4);
        Fraction fraction2 = new Fraction(2);
        Fraction fraction3 = new Fraction();
        System.out.print("Fraction 1: ");
        fraction1.displayFraction();
        System.out.print("Fraction 2: ");
        fraction2.displayFraction();
        System.out.print("Fraction 3: ");
        fraction3.displayFraction();
    }
}
```

```
Fraction 1: 3/4
Fraction 2: 2/1
Fraction 3: 1/1
```

8. Write a program to define a class Item containing code and price. Accept this data for five objects using array of objects. Display code, price in tabular form and also, display total price of all items.

Code:

```
public class Item {
    private int code;
    private double price;
    public Item(int code, double price) {
        this.code = code;
        this.price = price;
    }
    public int getCode() {
        return code;
    }
    public double getPrice() {
        return price;
    }
    public static void main(String[] args) {
        Item[] items = new Item[5];
        items[0] = new Item(101, 10.5);
        items[1] = new Item(102, 20.75);
        items[2] = new Item(103, 15.25);
        items[3] = new Item(104, 12.0);
        items[4] = new Item(105, 18.5);
        double totalPrice = 0.0;
        System.out.println("Code\tPrice");
        System.out.println("---------------");
        for (Item item : items) {
            System.out.println(item.getCode() + "\t$" + item.getPrice());
```

```
            totalPrice += item.getPrice();
        }
        System.out.println("---------------");
        System.out.println("Total Price: $" + totalPrice);
    }
}
```

```
Code    Price
---------------
101     $10.5
102     $20.75
103     $15.25
104     $12.0
105     $18.5
---------------
Total Price: $77.0
```

9. Write a program to define a class Tender containing data members cost and company name. Accept data for five objects and display company name for which cost is minimum.

```
import java.util.Scanner;
class Tender {
    private double cost;
    private String companyName;
    public Tender(double cost, String companyName) {
        this.cost = cost;
        this.companyName = companyName;    }
    public double getCost() {
        return cost;    }
    public String getCompanyName() {
        return companyName;    }}
public class Company {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Tender[] tenders = new Tender[5];
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter cost for tender " + (i + 1) + ": ");
            double cost = scanner.nextDouble();
            scanner.nextLine();
            System.out.println("Enter company name for tender " + (i + 1) + ": ");
            String companyName = scanner.nextLine();
            tenders[i] = new Tender(cost, companyName);       }
        double minCost = Double.MAX_VALUE;
        String minCompany = "";
        for (Tender tender : tenders) {
            if (tender.getCost() < minCost) {
                minCost = tender.getCost();
                minCompany = tender.getCompanyName();          }       }
```

```
    System.out.println("Company with minimum cost: " + minCompany);    }}
```

```
Enter cost for tender 1:
600
Enter company name for tender 1:
das company
Enter cost for tender 2:
4000
Enter company name for tender 2:
roy and das
Enter cost for tender 3:
4000
Enter company name for tender 3:
biswas and roy
Enter cost for tender 4:
8000
Enter company name for tender 4:
data consortium
Enter cost for tender 5:
2000
Enter company name for tender 5:
das pvt ltd
Company with minimum cost: das company
```

10. Write a program to define a class 'employee' with data members as empid, name and salary. Accept data for 5 objects using Array of objects and print it.

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Employee[] employees = new Employee[5];
        for (int i = 0; i < employees.length; i++) {
            employees[i] = new Employee();
            employees[i].acceptData();       }
        System.out.println("Employee Details:");
        for (int i = 0; i < employees.length; i++) {
            employees[i].displayData();         }    }}
class Employee {
    private int empid;
    private String name;
    private double salary;
    public void acceptData() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Employee ID:");
        empid = scanner.nextInt();
        System.out.println("Enter Employee Name:");
        name = scanner.next();
        System.out.println("Enter Employee Salary:");
        salary = scanner.nextDouble();    }
    public void displayData() {
        System.out.println("Employee ID: " + empid);
        System.out.println("Employee Name: " + name);
        System.out.println("Employee Salary: " + salary);
        System.out.println();    }}
```

```
Enter Employee ID:
1234
Enter Employee Name:
burnik
Enter Employee Salary:
6000
Enter Employee ID:
1235
Enter Employee Name:
arnab
Enter Employee Salary:
5000
Enter Employee ID:
1236
Enter Employee Name:
arpita
Enter Employee Salary:
7000
Enter Employee ID:
1237
Enter Employee Name:
antara
Enter Employee Salary:
2500
Enter Employee ID:
1238
Enter Employee Name:
rahul
Enter Employee Salary:
7900
```

```
Employee Details:
Employee ID: 1234
Employee Name: burnik
Employee Salary: 6000.0

Employee ID: 1235
Employee Name: arnab
Employee Salary: 5000.0

Employee ID: 1236
Employee Name: arpita
Employee Salary: 7000.0

Employee ID: 1237
Employee Name: antara
Employee Salary: 2500.0

Employee ID: 1238
Employee Name: rahul
Employee Salary: 7900.0
```

11. Define a class called circle that contains:
• Two private instance variables: radius (of type double) and color (of type String),
• Initialize the variables radius and color with default value of 1.0 and "red", respectively using default constructor.
• Include a second constructor that will use the default value for color and sets the
radius to the value passed as parameter.
• Two public methods: getRadius() and getArea() for returning the radius and area of
the circle
• Invoke the above methods and constructors in the main.
Code:

12. Write a program which will accept an integer from the user and pass the value to a method called PrintNumberInWord that will print "ONE", "TWO",... , "NINE", "ZERO" if the integer variable "number" is 1, 2,... , 9, or 0, respectively.

```
import java.util.Scanner;
```

```
public class NWC {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter an integer between 0 and 9: ");
      int number = scanner.nextInt();
      scanner.close();
      printNumberInWord(number);    }
      public static void printNumberInWord(int number) {
      switch (number) {
         case 0:
            System.out.println("ZERO");
            break;
         case 1:
            System.out.println("ONE");
            break;
         case 2:
            System.out.println("TWO");
            break;
         case 3:
            System.out.println("THREE");
            break;
         case 4:
            System.out.println("FOUR");
            break;
         case 5:
            System.out.println("FIVE");
            break;
         case 6:
            System.out.println("SIX");
            break;
         case 7:
            System.out.println("SEVEN");
            break;
         case 8:
            System.out.println("EIGHT");
            break;
         case 9:
            System.out.println("NINE");
            break;
         default:
            System.out.println("Number out of range.");       }   }}
Enter an integer between 0 and 9: 6
SIX
PS C:\Users\burni\OneDrive\Desktop\java>
```

13. Design a class named Account that contains:
I. A private int data field named id for the account (default 0).
II. A private double data field named balance for the account (default 0).
III. A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.

IV. A private Date data field named dateCreated that stores the date when the account was created.
V. A no-arg constructor that creates a default account.
VI. A constructor that creates an account with the specified id and initial balance.
VII. The accessor and mutator methods for id,balance, and annualInterestRate.
VIII. The accessor method for dateCreated.
IX. A method named getMonthlyInterestRate() that returns the monthly interest rate.
X. A method named getMonthlyInterest() that returns the monthly interest.
XI. A method named withdraw that withdraws a specified amount from the account.
XII. A method named deposit that deposits a specified amount to the account.
Code:

```
import java.util.Date;
public class Account {
    private int id;
    private double balance;
    private double annualInterestRate;
    private Date dateCreated;
    public Account() {
        id = 0;
        balance = 0;
        annualInterestRate = 0;
        dateCreated = new Date();
    }
    public Account(int id, double balance) {
        this.id = id;
        this.balance = balance;
        annualInterestRate = 0;
        dateCreated = new Date();
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }
    public double getAnnualInterestRate() {
        return annualInterestRate;
    }
    public void setAnnualInterestRate(double annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
    }
```

```
    public Date getDateCreated() {
        return dateCreated;
    }
    public double getMonthlyInterestRate() {
        return annualInterestRate / 12;
    }
    public double getMonthlyInterest() {
        return balance * (getMonthlyInterestRate() / 100);
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful: $" + amount);
        } else {
            System.out.println("Withdrawal failed: Insufficient funds");
        }
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful: $" + amount);
        } else {
            System.out.println("Deposit failed: Invalid amount");
        }
    }
    public static void main(String[] args) {
        Account account1 = new Account();
        account1.setId(1001);
        account1.setBalance(5000);
        account1.setAnnualInterestRate(3.5);
        System.out.println("Account ID: " + account1.getId());
        System.out.println("Balance: $" + account1.getBalance());
        System.out.println("Annual Interest Rate: " + account1.getAnnualInterestRate()
+ "%");
        System.out.println("Date Created: " + account1.getDateCreated());
        System.out.println("Monthly Interest Rate: " +
account1.getMonthlyInterestRate() + "%");
        System.out.println("Monthly Interest: $" + account1.getMonthlyInterest());
        account1.withdraw(1000);
        System.out.println("Current Balance after withdrawal: $" +
account1.getBalance());
        account1.deposit(2000);
        System.out.println("Current Balance after deposit: $" + account1.getBalance());
    }
}
```

```
Account ID: 1001
Balance: $5000.0
Annual Interest Rate: 3.5%
Date Created: Wed Feb 21 12:10:40 IST 2024
Monthly Interest Rate: 0.2916666666666667%
Monthly Interest: $14.583333333333334
Withdrawal successful: $1000.0
Current Balance after withdrawal: $4000.0
Deposit successful: $2000.0
Current Balance after deposit: $6000.0
PS D:\JAVA Lab Assignments>
```

14. Write a test program that prompts the user to enter the investment amount (e.g., 1000) and the interest rate (e.g., 9%), and print a table that displays future value for the years from 1 to 30, as shown below:

The amount invested: 1000
Annual interest rate: 9%
Years Future Value
1 1093.8
2 1196.41
...
29 13467.25
30 14730.57

15. Write method headers for the following methods:

- **Computing a sales commission, given the sales amount and the commission rate.**

```
public double computeSalesCommission(double salesAmount, double
commissionRate) {
   // Method body
}
```

- **Printing the calendar for a month, given the month and year.**

```
public void printCalendar(int month, int year) {

}
```

- **Computing a square root.**

```
public double computeSquareRoot(double number) {
}
```

- **Testing whether a number is even, and returning true if it is.**

```
public boolean isEven(int number) {

}
```

- **Printing a message a specified number of times.**

```
public void printMessage(String message, int times) {
}
```

- **Computing the monthly payment, given the loan amount, number of years, and annual interest rate.**

```
public double computeMonthlyPayment(double loanAmount, int
numberOfYears, double annualInterestRate) {

}
```

16. Write a program that reads ten numbers, computes their average, and finds out how many numbers are above the average. [Use this keyword]

```
import java.util.Scanner;
public class AverageAndAbove {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      double[] numbers = new double[10];
      System.out.println("Enter ten numbers:");
      for (int i = 0; i < 10; i++) {
         numbers[i] = scanner.nextDouble();       }
      double sum = 0;
      for (double num : numbers) {
         sum += num;       }
      double average = sum / 10;
      int aboveAverageCount = 0;
      for (double num : numbers) {
         if (num > average) {
            aboveAverageCount++;          }      }
      System.out.println("Average: " + average);
    System.out.println("Number of elements above the average: " + aboveAverageCount);}}
```

```
Enter ten numbers:
23
45
40
67
80
60
65
25
35
70
Average: 51.0
Number of elements above the average: 5
```

17. Write a program that reads ten integers and displays them in the reverse of the order in which they were read.

```
import java.util.Scanner;
public class ReverseOrder {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] numbers = new int[10];
        System.out.println("Enter ten integers:");
        for (int i = 0; i < 10; i++) {
            numbers[i] = scanner.nextInt();        }
        System.out.println("Integers in reverse order:");
        for (int i = 9; i >= 0; i--) {
```

System.out.println(numbers[i]); } }}

```
Enter ten integers:
23
45
67
43
35
87
65
46
79
70
Integers in reverse order:
70
79
46
65
87
35
43
67
45
23
```

18. Write a program to demonstrate use of 'this' keyword.

Code: public class Person {

```
    private String name;
    public Person(String name) {
        this.name = name;
    }
    public void display() {
        System.out.println("Name: " + this.name);
    }
    public void setName(String name) {
        this.name = name;
    }
    public static void main(String[] args) {
        Person person1 = new Person("Rahul");
        System.out.println("Name before update:");
        person1.display();
        person1.setName("Raj");
        System.out.println("\nName after update:");
        person1.display();
    }
}
```

```
Name before update:
Name: Rahul
Name after update:
Name: Raj
PS D:\JAVA_Lab_Assignments>
```

19. Write a program to demonstrate use of 'static' keyword.

Code: public class Counter {

```
    private static int count = 0;
```

```
    public Counter() {
        count++;
    }
    public static int getCount() {
        return count;
    }
    public static void main(String[] args) {
        Counter obj1 = new Counter();
        Counter obj2 = new Counter();
        Counter obj3 = new Counter();
        System.out.println("Number of objects created: " + Counter.getCount());
    }
}
```

```
Number of objects created: 3
```

20. Write a program to accept value of apple sales for each day of the week (using array of type float) and then, calculate the average sale of the week.

```
import java.util.Scanner;
public class WeeklyAppleSales {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        float[] sales = new float[7];
        for (int i = 0; i < 7; i++) {
            System.out.print("Enter sales value for day " + (i + 1) + ": ");
            sales[i] = scanner.nextFloat();        }
        float sum = 0;
        for (float sale : sales) {
            sum += sale;
        }
        float average = sum / 7;
        System.out.println("Average sale for the week: " + average);    }}
```

```
Enter sales value for day 1: 24
Enter sales value for day 2: 45
Enter sales value for day 3: 67
Enter sales value for day 4: 50
Enter sales value for day 5: 100
Enter sales value for day 6: 68
Enter sales value for day 7: 70
Average sale for the week: 60.57143
```

21. Write program, which finds the sum of numbers formed by consecutive digits. Input : 2415 output : 24+41+15=80.

```
import java.util.Scanner;
public class ConsecutiveDigitSum {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        String input = scanner.nextLine();
        int sum = findConsecutiveDigitSum(input);
```

```
System.out.println("Sum of numbers formed by consecutive digits: " + sum);
}   public static int findConsecutiveDigitSum(String input) {
        int sum = 0;
        for (int i = 0; i < input.length() - 1; i++) {
            int currentDigit = Character.getNumericValue(input.charAt(i));
            int nextDigit = Character.getNumericValue(input.charAt(i + 1));
            int number = currentDigit * 10 + nextDigit;
            sum += number;        }
        return sum;    }}
```

```
Enter a number: 2415
Sum of numbers formed by consecutive digits: 80
```