Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

BHIMTAL CAMPUS

PRACTICAL FILE / TERM WORK

SUBJECT NAME: Java Programming Laboratory

SUBJECT CODE: PBC 501

COURSE – Bachelor of Computer Applications

SEMESTER – 5th

SESSION: 2023-2026

SUBMITTED TO	SUBMITTED BY
Concerned Faculty	Name- Karan Suyal
Mr. Devendra Singh	University Roll NO- 2371172

Graphic Era

HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

BHIMTAL CAMPUS

CERTIFICATE

This is to certify that **Karan Suyal** has satisfactorily completed all the experiment in the laboratory of this college. The Term Work of

Java Programming Laboratory. In partial fulfilment of the requirement in **5th semester** of **Bachelor of Computer Applications** degree course prescribed by **Graphic Era Hill University, Bhimtal.**

Concerned Faculty	**HEAD OF DEPARTMENT**
Mr**.** Devendra Singh	DR. Sandeep Kumar Budhani

ACKNOWLEDGEMENT

I express my gratitude to teachers, for coming up with this interesting and thought- provoking course, compiling the best contents and the most important working towards changing the words, making the differences and inspiring us to become the climate saver ourselves, knowledge that I gained will last a lifetime.

Many thanks to my institute, **Graphic Era Hill University, Bhimtal** and a special thanks to our Concerned Faculty **Mr. Devendra Singh** for motivating us to look beyond the syllabus, to take the opportunity and to learn as much as we can. I also thank Mam for helping us in this term work.

Student Name –Karan Suyal

University Roll NO.: 2371172

Graphic Era

HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

BHIMTAL CAMPUS

Index

10.	Develop a Java program to manipulate a string using methods of the String class (e.g., length(), to Upper Case(), substring()). Take a string as user input and demonstrate at least three string operations.	10	
11.	Write a Java program to reverse a string using the String Buffer class. Take the string as user input and display the reversed string.	11	
12.	Create a Java program to demonstrate a nested for loop to print a multiplication table (1 to 10) in a tabular format.	12	
13.	Write a Java program to create a class Bank Account with private attributes account Number, balance, and account Holder Name. Implement methods for deposit, withdrawal, and balance inquiry using appropriate visibility labels.	13-14	
14.	Develop a Java program to demonstrate inheritance by creating a superclass Vehicle with attributes brand and year, and a subclass Car with an additional attribute model. Create objects of both classes and display their details.	15-16	
15.	Create a Java program to demonstrate method overloading by defining multiple calculate Area methods in a class Shape to compute the area of a circle, rectangle, and triangle based on different parameters.	17	
16.	Write a Java program to create an abstract class Animal with an abstract method make Sound(). Implement two subclasses Dog and Cat that override the make Sound() method to print appropriate sounds.	18	
17.	Develop a Java program to demonstrate multithreading by creating two threads using the Thread class. One thread prints even numbers from 1 to 20, and the other prints odd numbers. Ensure proper synchronization to avoid interleaved output.	19	
18.	Create a Java program to demonstrate the use of an interface Resizable with a method resize(int percentage). Implement this interface in a class Image to simulate resizing an image by updating its dimensions.	20	
19.	Write a Java program to create a package mathutils containing a class Calculator with static methods for addition, subtraction, and multiplication. Import and use this package in another class to perform calculations.	21	
20.	Develop a Java program to demonstrate dynamic method dispatch using a superclass Shape with a method draw() and two subclasses Circle and Square that override the draw() method. Use a superclass reference to call the overridden methods.	22	

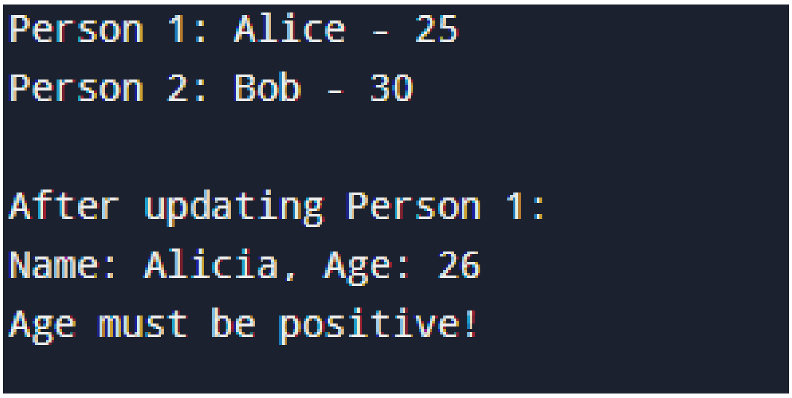
Q1-Write a Java program to create a class Person with attributes name and age. Implement encapsulation using private variables and public getter/setter methods. Demonstrate creating and accessing objects of this class in the main method.

Code-

```
class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        if (age > 0) {
            this.age = age;
        } else {
            System.out.println("Age must be positive!");
        }
    }
    public void displayInfo() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Person p1 = new Person("Alice", 25);
        Person p2 = new Person("Bob", 30);
        System.out.println("Person 1: " + p1.getName() + " - " + p1.getAge());
        System.out.println("Person 2: " + p2.getName() + " - " + p2.getAge());
        p1.setName("Alicia");
        p1.setAge(26);
        System.out.println("\nAfter updating Person 1:");
        p1.displayInfo();
        p2.setAge(-5);
    }
}
```

Output-



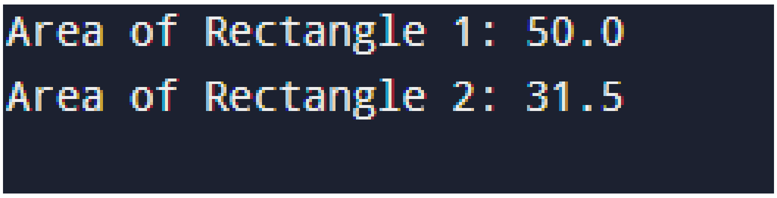
```
Person 1: Alice - 25
Person 2: Bob - 30

After updating Person 1:
Name: Alicia, Age: 26
Age must be positive!
```

Q2- Create a Java program to calculate the area of a rectangle using a class Rectangle with private attributes length and width. Use a constructor to initialize the attributes and a method to compute the area.

Code-

```
class Rectangle {

    private double length;

    private double width;

    public Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

    public double calculateArea() {

        return length * width;

    }

}

public class Main {

    public static void main(String[] args) {

        Rectangle rect1 = new Rectangle(10, 5);

        Rectangle rect2 = new Rectangle(7.5, 4.2);

        System.out.println("Area of Rectangle 1: " + rect1.calculateArea());

        System.out.println("Area of Rectangle 2: " + rect2.calculateArea());

    }

}
```

Output-



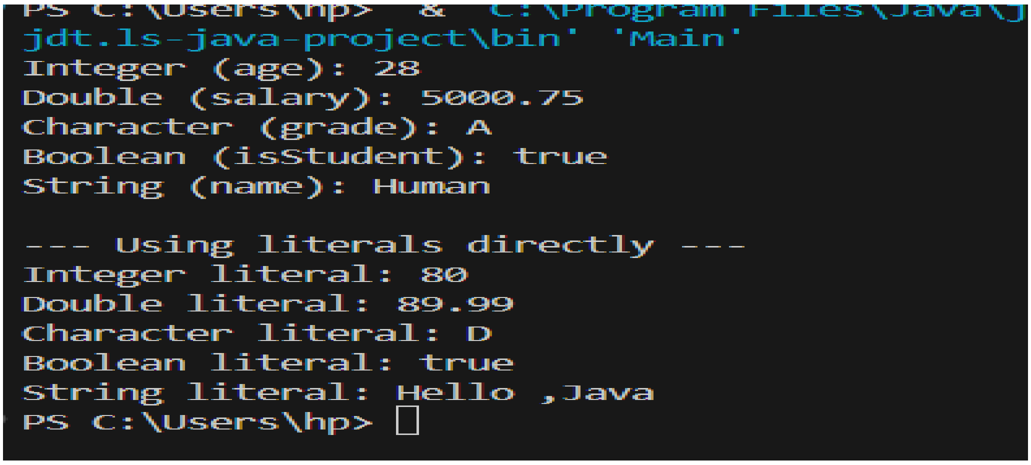
```
Area of Rectangle 1: 50.0
Area of Rectangle 2: 31.5
```

Q3-Write a Java program to demonstrate the use of different data types (int, double, char, boolean, String) and display their values using variables and literals.

Code-

```
public class Main {
    public static void main(String[] args) {
        int age = 28;
        double salary = 5000.75; char
        grade = 'A';
        boolean isStudent = true;
        String name = "Human";
        System.out.println("Integer (age): " + age);
        System.out.println("Double (salary): " + salary);
        System.out.println("Character (grade): " + grade);
        System.out.println("Boolean (isStudent): " + isStudent);
        System.out.println("String (name): " + name);
        System.out.println("\n--- Using literals directly ---");
        System.out.println("Integer literal: " + 80);
        System.out.println("Double literal: " + 89.99);
        System.out.println("Character literal: " + 'D');
        System.out.println("Boolean literal: " + true);
        System.out.println("String literal: " + "Hello ,Java");
    }
}
```

Output-



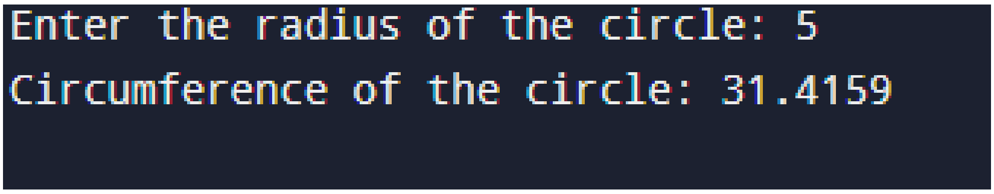
```
PS C:\Users\hp>  & C:\Program Files\Java\j
jdt.ls-java-project\bin' 'Main'
Integer (age): 28
Double (salary): 5000.75
Character (grade): A
Boolean (isStudent): true
String (name): Human

--- Using literals directly ---
Integer literal: 80
Double literal: 89.99
Character literal: D
Boolean literal: true
String literal: Hello ,Java
PS C:\Users\hp> 
```

Q4- Develop a Java program that declares constants using the final keyword to store the value of PI and use it to calculate the circumference of a circle given the radius as user input

Code-

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        final double PI = 3.14159;

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the radius of the circle: ");

        double radius = sc.nextDouble();

        double circumference = 2 * PI * radius;

        System.out.println("Circumference of the circle: " + circumference);

        sc.close();

    }

}
```

Output:



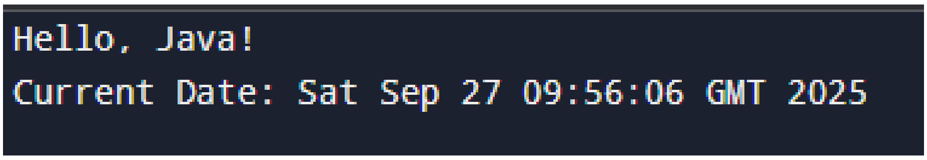
```
Enter the radius of the circle: 5
Circumference of the circle: 31.4159
```

Q5- Create a Java program to demonstrate the structure of a Java program, including package declaration, import statements, class definition, and the main method. Print "Hello, Java!" as output.

Code-

```
package  mypackage;

import java.util.Date;

public class Main {

    public static void main(String[] args) {

        System.out.println("Hello,  Java!");

        Date today = new Date();

        System.out.println("Current Date: " + today);

    }

}
```

Output-



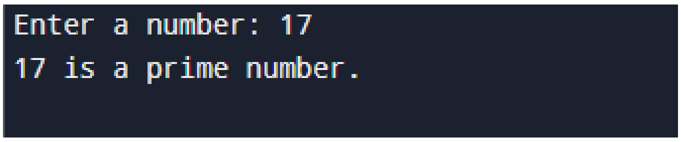
```
Hello, Java!
Current Date: Sat Sep 27 09:56:06 GMT 2025
```

Q6- Write a Java program to check if a given number is prime using a for loop and if-else statements. Take the number as user input

Code-

```
import java.util.Scanner;

public class PrimeCheck {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int num = sc.nextInt();

        boolean isPrime = true;

        if (num <= 1) {

            isPrime = false;

        } else {

            for (int i = 2; i <= num / 2; i++) {

                if (num % i == 0) {

                    isPrime = false;

                    break;

                }

            }

        }

        if (isPrime) {

            System.out.println(num + " is a prime number.");

        } else {

            System.out.println(num + " is not a prime number.");

        }

        sc.close();

    }}
```

Output-



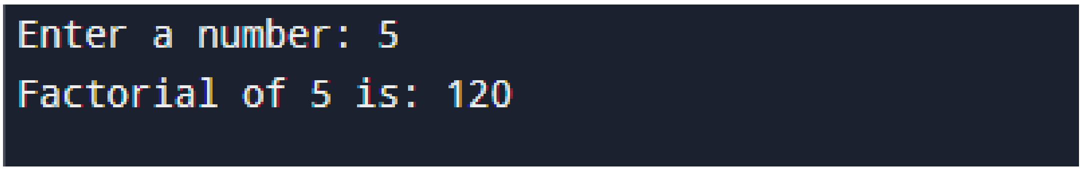
```
Enter a number: 17
17 is a prime number.
```

Q7- Develop a Java program to find the factorial of a number using a while loop. Handle invalid input using if-else.

Code-

```
import java.util.Scanner;

public class FactorialWhile {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int num = sc.nextInt();

        if (num < 0) {

            System.out.println("Invalid input! Factorial is not defined for negative numbers.");

        } else {

            long fact = 1;

            int i = 1;

            while (i <= num) {

                fact *= i;

                i++;

            }

            System.out.println("Factorial of " + num + " is: " + fact);

        }

        sc.close();

    }

}
```

Output-



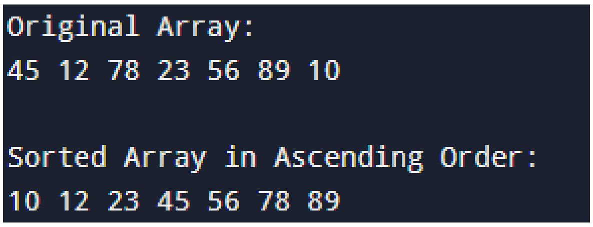
```
Enter a number: 5
Factorial of 5 is: 120
```

Q8- Create a Java program to initialize a one-dimensional array of integers, sort it in ascending order using the bubble sort algorithm, and display the sorted array.

Code-

```
public class BubbleSortExample {
    public static void main(String[] args) {
        int[] arr = {45, 12, 78, 23, 56, 89, 10};
        System.out.println("Original Array:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
        // Bubble Sort Algorithm
        for (int i = 0; i < arr.length - 1; i++) {
            for (int j = 0; j < arr.length - 1 - i; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        System.out.println("\n\nSorted Array in Ascending Order:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }
}
```

Output-



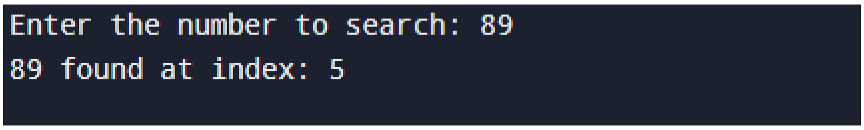
```
Original Array:
45 12 78 23 56 89 10

Sorted Array in Ascending Order:
10 12 23 45 56 78 89
```

Q9-Write a Java program to perform a binary search on a sorted array of integers. Take user input for the element to search and display its index or a message if not found.

Code-

```
import java.util.Scanner;
public class BinarySearchExample {
  public static void main(String[] args) {
    int[] arr = {10, 23, 45, 56, 78, 89, 95};
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the number to search: ");
    int key = sc.nextInt();
    int low = 0;
    int high = arr.length - 1;
    int index = -1;
    while (low <= high) {
      int mid = (low + high) / 2;
      if (arr[mid] == key) {
        index = mid;
        break;
      } else if (arr[mid] < key) {
        low = mid + 1;
      } else {
        high = mid - 1;
      } }
    if (index != -1) {
      System.out.println(key + " found at index: " + index);
    } else {
      System.out.println(key + " not found in the array.");
    }sc.close();
  }}
```

Output-



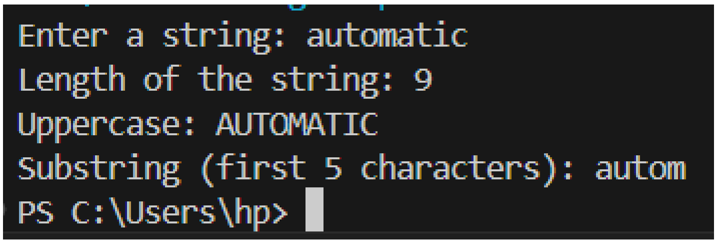
```
Enter the number to search: 89
89 found at index: 5
```

Q10-Develop a Java program to manipulate a string using methods of the String class (e.g., length(), to Upper Case(), substring()). Take a string as user input and demonstrate at least three string operations.

Code-

```
import java.util.Scanner;

public class StringManipulation {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        String input = sc.nextLine();

        int length = input.length();

        System.out.println("Length of the string: " + length);

        String upper = input.toUpperCase();

        System.out.println("Uppercase: " + upper);

        String sub = input.length() >= 5 ? input.substring(0, 5) : input;
        System.out.println("Substring (first 5 characters): " + sub)
        sc.close();

    }

}
```

Output-



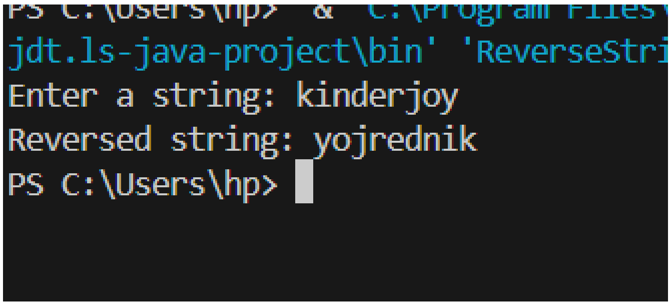
```
Enter a string: automatic
Length of the string: 9
Uppercase: AUTOMATIC
Substring (first 5 characters): autom
PS C:\Users\hp>
```

Q11-Write a Java program to reverse a string using the String Buffer class. Take the string as user input and display the reversed string

Code-

```
import java.util.Scanner;

public class ReverseString {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        String input = sc.nextLine();

        StringBuffer sb = new StringBuffer(input);

        sb.reverse();

        System.out.println("Reversed string: " + sb.toString());

        sc.close();

    }

}
```

Output-



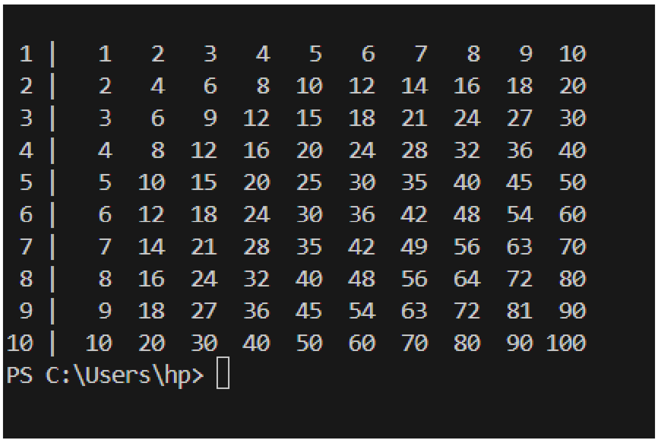
```
PS C:\Users\hp>  & 'C:\Program Files\
jdt.ls-java-project\bin' 'ReverseStri
Enter a string: kinderjoy
Reversed string: yojrednik
PS C:\Users\hp> 
```

Q12-Create a Java program to demonstrate a nested for loop to print a multiplication table (1 to 10) in a tabular format.

Code-

```
public class MultiplicationTable {
    public static void main(String[] args) {
        // Print table header
        System.out.print("    ");
        for (int i = 1; i <= 10; i++) {
            System.out.printf("%4d",  i);
        }
        System.out.println("\n---------------------------------------------");
        for (int i = 1; i <= 10; i++) {
        System.out.printf("%2d |", i); // Row label for
        (int j = 1; j <= 10; j++) {
                System.out.printf("%4d", i * j);
            }
            System.out.println();
        }
    }}
```

Output-



```
 1 |   1   2   3   4   5   6   7   8   9  10
 2 |   2   4   6   8  10  12  14  16  18  20
 3 |   3   6   9  12  15  18  21  24  27  30
 4 |   4   8  12  16  20  24  28  32  36  40
 5 |   5  10  15  20  25  30  35  40  45  50
 6 |   6  12  18  24  30  36  42  48  54  60
 7 |   7  14  21  28  35  42  49  56  63  70
 8 |   8  16  24  32  40  48  56  64  72  80
 9 |   9  18  27  36  45  54  63  72  81  90
10 |  10  20  30  40  50  60  70  80  90 100
PS C:\Users\hp> 
```

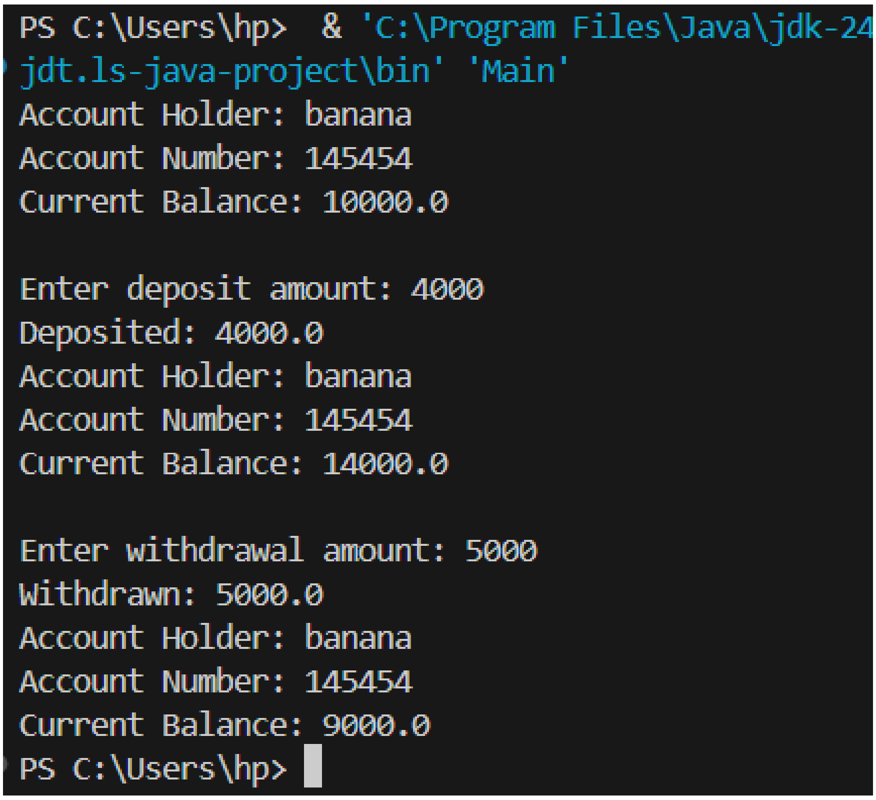
Q13-Write a Java program to create a class Bank Account with private attributes account Number, balance, and account Holder Name. Implement methods for deposit, withdrawal, and balance inquiry using appropriate visibility labels

Code-

```
import java.util.Scanner;


class BankAccount {
  private String accountNumber;
  private String accountHolderName;
  private double balance;


  public BankAccount(String accountNumber, String accountHolderName, double initialBalance) {
    this.accountNumber = accountNumber;
    this.accountHolderName = accountHolderName;
    this.balance = initialBalance;
  }


  public void deposit(double amount) {
    if (amount > 0) {
      balance += amount;
      System.out.println("Deposited: " + amount);
    } else {
      System.out.println("Invalid deposit amount.");
    }
  }


  public void withdraw(double amount) {
    if (amount > 0) {
      if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
```

```
            } else {
                System.out.println("Insufficient balance.");
            }
        } else {
            System.out.println("Invalid withdrawal amount.");
        }
    }

    public void displayBalance() {
        System.out.println("Account Holder: " + accountHolderName);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Current Balance: " + balance);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        BankAccount account = new BankAccount("1234567890", "Divyanshu Mehta", 5000);
        account.displayBalance();
        System.out.println();
        System.out.print("Enter deposit amount: ");

        double depositAmount = sc.nextDouble();
        account.deposit(depositAmount);
        account.displayBalance();

        System.out.println();
        System.out.print("Enter withdrawal amount: ");
        double withdrawalAmount = sc.nextDouble();
        account.withdraw(withdrawalAmount);

        account.displayBalance();

        sc.close();

    }

}
```

Output-



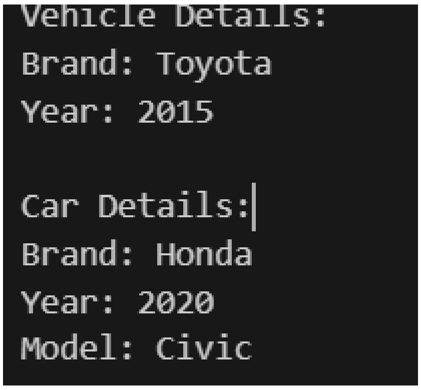
```
PS C:\Users\hp>  & 'C:\Program Files\Java\jdk-24
jdt.ls-java-project\bin' 'Main'
Account Holder: banana
Account Number: 145454
Current Balance: 10000.0

Enter deposit amount: 4000
Deposited: 4000.0
Account Holder: banana
Account Number: 145454
Current Balance: 14000.0

Enter withdrawal amount: 5000
Withdrawn: 5000.0
Account Holder: banana
Account Number: 145454
Current Balance: 9000.0
PS C:\Users\hp>
```

Q14-Develop a Java program to demonstrate inheritance by creating a superclass Vehicle with attributes brand and year, and a subclass Car with an additional attribute model. Create objects of both classes and display their details.

Code-

```
class Vehicle {
    protected String brand;
    protected int year;

    public Vehicle(String brand, int year) {
        this.brand = brand;
        this.year = year;
    }

    public void displayVehicleInfo() {
        System.out.println("Brand: " + brand);
        System.out.println("Year: " + year);
    }
}

class Car extends Vehicle {
    private String model;

    public Car(String brand, int year, String model) {
        super(brand, year);
        this.model = model;
    }

    public void displayCarInfo() {
        displayVehicleInfo();
        System.out.println("Model: " + model);
    }
}
public class Main {
    public static void main(String[] args) {
        Vehicle vehicle = new Vehicle("Toyota", 2015);
        Car car = new Car("Honda", 2020, "Civic");
        System.out.println("Vehicle Details:");
        vehicle.displayVehicleInfo();
        System.out.println("\nCar Details:");
        car.displayCarInfo();
    }
}
```

Output:



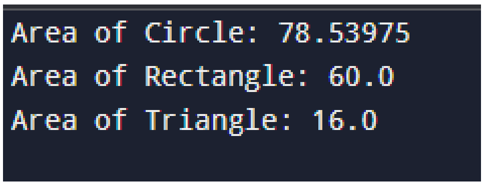
```
Vehicle Details:
Brand: Toyota
Year: 2015

Car Details:
Brand: Honda
Year: 2020
Model: Civic
```

Q15-Create a Java program to demonstrate method overloading by defining multiple calculate Area methods in a class Shape to compute the area of a circle, rectangle, and triangle based on different parameters.

Code-

```
class Shape {
  public double calculateArea(double radius) {
    return 3.14159 * radius * radius;
  }
  public double calculateArea(double length, double width) {
    return length * width;
  }
  public double calculateArea(double base, double height, boolean isTriangle) {
    return 0.5 * base * height;
  }
}
public class Main {
  public static void main(String[] args) {
    Shape shape = new Shape();
    double circleArea = shape.calculateArea(5);
    double rectangleArea = shape.calculateArea(10, 6);
    double triangleArea = shape.calculateArea(8, 4, true);

    System.out.println("Area of Circle: " + circleArea);
    System.out.println("Area of Rectangle: " + rectangleArea);
    System.out.println("Area of Triangle: " + triangleArea);
  }
}
```

Output-



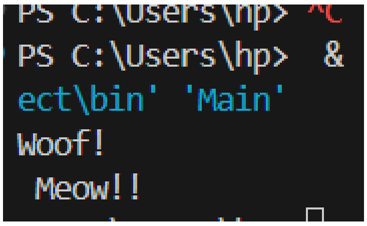
```
Area of Circle: 78.53975
Area of Rectangle: 60.0
Area of Triangle: 16.0
```

Q16-Write a Java program to create an abstract class Animal with an abstract method make Sound(). Implement two subclasses Dog and Cat that override the make Sound() method to print appropriate sounds.

Code-

```
abstract class Animal {
    public abstract void makeSound();
}
class Dog extends Animal {
    public void makeSound()
    {

        System.out.println("Woof!");
    }
}
class Cat extends Animal {
    @Override
    public void makeSound() {
        System.out.println(" Meow!!");
    }
}
public class Main {
    public static void main(String[] args) {
        Animal dog = new Dog();
        Animal cat = new Cat();
        dog.makeSound();
        cat.makeSound();
    }
}
```

Output-



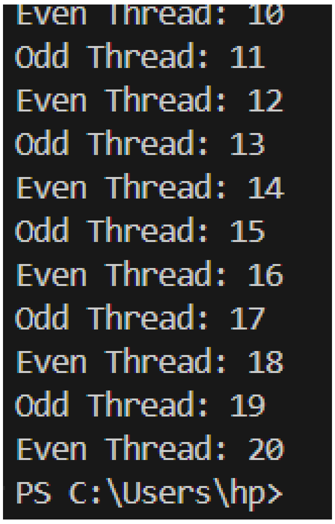
```
PS C:\Users\hp> ^C
PS C:\Users\hp>  &
ect\bin' 'Main'
Woof!
 Meow!!
```

Q17-Develop a Java program to demonstrate multithreading by creating two threads using the Thread class. One thread prints even numbers from 1 to 20, and the other prints odd numbers. Ensure proper synchronization to avoid interleaved output

Code-

```
class NumberPrinter {
    private int number = 1;
    private final int MAX = 20;
    public synchronized void printOdd() {
        while (number <= MAX) {
            while (number % 2 == 0) {
                try {
                    wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            if (number <= MAX) {
                System.out.println("Odd Thread: " + number);
                number++;
                notify();
            }
        }
    }  public synchronized void printEven() {
        while (number <= MAX) {
            while (number % 2 == 1) {
                try {
                    wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            if (number <= MAX) {
                System.out.println("Even Thread: " + number);
                number++;
                notify();
            }
        }
    }
}

class OddThread extends Thread {
    private NumberPrinter printer;
    public OddThread(NumberPrinter printer) {
        this.printer = printer;
    }
    public void run() {
        printer.printOdd();
    }
}

class EvenThread extends Thread {
    private NumberPrinter printer;
    public EvenThread(NumberPrinter printer) {
        this.printer = printer;
    }
    public void run() {
        printer.printEven();
    }
}
public class EvenOddSync {
    public static void main(String[] args) {
        NumberPrinter printer = new NumberPrinter();
        Thread t1 = new OddThread(printer);
        Thread t2 = new EvenThread(printer);
        t1.start();
        t2.start();
    }
}
```

Output:



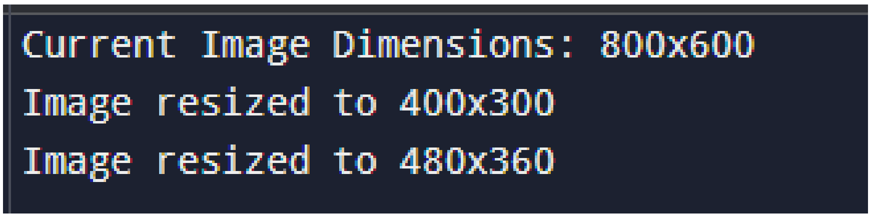
```
Even Thread: 10
Odd Thread: 11
Even Thread: 12
Odd Thread: 13
Even Thread: 14
Odd Thread: 15
Even Thread: 16
Odd Thread: 17
Even Thread: 18
Odd Thread: 19
Even Thread: 20
PS C:\Users\hp>
```

Q18-Create a Java program to demonstrate the use of an interface Resizable with a method resize(int percentage). Implement this interface in a class Image to simulate resizing an image by updating its dimensions.

Code-

```
interface Resizable {
    void resize(int percentage);
}
class Image implements Resizable {
    private int width;
    private int height;
    public Image(int width, int height) {
        this.width = width;
        this.height = height;
    }
    @Override
    public void resize(int percentage) {
        if (percentage > 0) {
            width = width * percentage / 100;
            height = height * percentage / 100;
            System.out.println("Image resized to " + width + "x" + height);
        } else {
            System.out.println("Invalid resize percentage!");
        }
    }
    public void displayDimensions() {
        System.out.println("Current Image Dimensions: " + width + "x" + height);
    }
}
public class Main {
    public static void main(String[] args) {
        Image img = new Image(800, 600);
        img.displayDimensions();


        img.resize(50);

        img.resize(120);

    }

}
```

Output-



```
Current Image Dimensions: 800x600
Image resized to 400x300
Image resized to 480x360
```

Q19-Write a Java program to create a package mathutils containing a class Calculator with static methods for addition, subtraction, and multiplication. Import and use this package in another class to perform calculations.

Code-

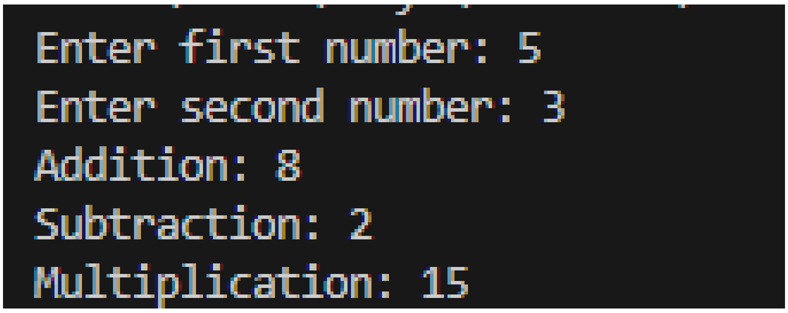
Calculator.java

```
package mathutils;

public class Calculator {

    public static int add(int a, int b) {

        return a + b;

    }

    public static int subtract(int a, int b) {

        return a - b;

    }

    public static int multiply(int a, int b) {

        return a * b;

    }

}
```

Main.java

```
import mathutils.Calculator;

import java.util.Scanner;


public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);


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

        int num1 = sc.nextInt();


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

        int num2 = sc.nextInt();
        System.out.println("Addition: " + Calculator.add(num1, num2));

        System.out.println("Subtraction: " + Calculator.subtract(num1, num2));

        System.out.println("Multiplication: " + Calculator.multiply(num1, num2));


        sc.close();

    }

}
```

Output-



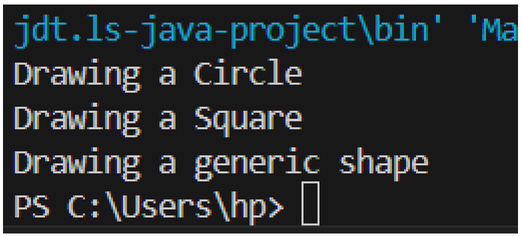
```
Enter first number: 5
Enter second number: 3
Addition: 8
Subtraction: 2
Multiplication: 15
```

Q20-Develop a Java program to demonstrate dynamic method dispatch using a superclass Shape with a method draw() and two subclasses Circle and Square that override the draw() method. Use a superclass reference to call the overridden methods.

<u>Code-</u>

```
class Shape {
    public void draw() {
        System.out.println("Drawing a generic shape");
    }}
class Circle extends Shape {
   public void draw() {
        System.out.println("Drawing a Circle");
    }}
class Square extends Shape {
  public void draw() {
        System.out.println("Drawing a Square");
    }}
public class Main {
    public static void main(String[] args) {
        Shape shape;
        shape = new Circle();
        shape.draw();

         shape = new Square();
        shape.draw();
        shape = new Shape();
        shape.draw();
    }}
```

<u>Output-</u>



```
jdt.ls-java-project\bin' 'Ma
Drawing a Circle
Drawing a Square
Drawing a generic shape
PS C:\Users\hp> 
```