Task 1

public class task1 {

    public static void main(String[] args) {

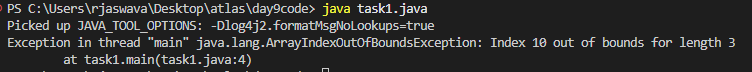
        int[] myNumbers = {1, 2, 3};

        System.out.println(myNumbers[10]);

    }

}

Output



Task 2

public class task2 {

    public static void main(String[] args) {

        try {

            int[] myNumbers = {1, 2, 3};

            System.out.println(myNumbers[10]);

        } catch (Exception e) {

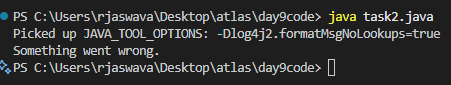
            System.out.println("Something went wrong.");

        }

    }

}

Output



Task 3

public class task3 {

    public static void main(String[] args) {

        try {

            int[] myNumbers = {1, 2, 3};

            System.out.println(myNumbers[10]);

        } catch (Exception e) {

            System.out.println("Something went wrong.");

        } finally {

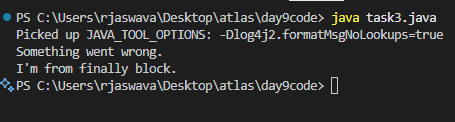
            System.out.println("I'm from finally block.");

        }

    }

}

Output



Task 4

package day9code;

public class task4 {

    public static void main(String[] args) {

        try {

            int[] myNumbers = {1, 2, 3};

            System.out.println(myNumbers[1]);

        } catch (Exception e) {

            System.out.println("Something went wrong.");

        } finally {

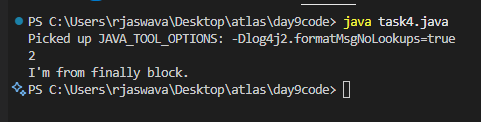
            System.out.println("I'm from finally block.");

        }

    }

}

Output



Task 5

public class task5 {

    public static void main(String args[]) {

        try {

            int a[] = new int[2];

            int b = 0;

            int c = 1/b;

            System.out.println("Access element three :" + a[3]);

        }

        catch (ArrayIndexOutOfBoundsException e) {

            System.out.println("ArrayIndexOutOfBoundsException thrown  :" + e);

        }catch (Exception e) {

            System.out.println("Exception thrown  :" + e);

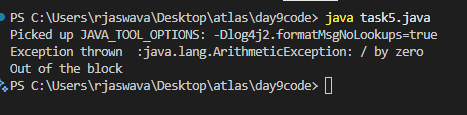
        }

        System.out.println("Out of the block");

    }

}

Output



Task 6

public class task6 {

    public static void main(String args[]) {

        try {

            int a[] = new int[2];

            System.out.println("Access element three :" + a[0]);

            try {

                int b = 0;

                int c = 1/b;

            }catch(Exception e) {

                System.out.println("Exception thrown: " + e);

            }

            System.out.println("Access element three :" + a[3]);

        }

        catch (ArrayIndexOutOfBoundsException e) {

            System.out.println("Exception thrown: " + e);

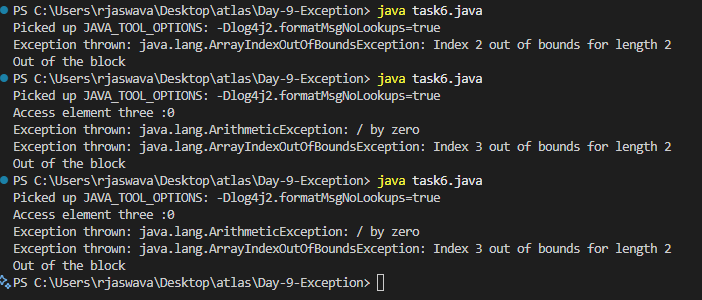
        }

        System.out.println("Out of the block");

    }

}

Output



Task 7

public class task7 {

    static void fun() throws IllegalAccessException {

        System.out.println("Inside fun(). ");

        throw new IllegalAccessException("demo exception by fun method");

    }

    public static void main(String args[]) {

        try {

            fun();

        }

        catch (IllegalAccessException e) {

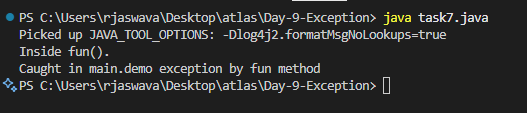
            System.out.println("Caught in main." + e.getMessage());

        }

    }

}

Output



Task 8

class MyException extends Exception {

    public MyException(String m) {

        super(m);

    }

}

public class task8 {

    public static void main(String args[]) {

        try {

            throw new MyException("This is a custom exception");

        }

        catch (MyException ex) {

            System.out.println("Caught");

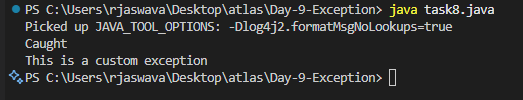
            System.out.println(ex.getMessage());

        }

    }

}

Output



Task 9

import java.util.ArrayList;

public class task9 {

    public static void main(String[] args) {

        // Creating an ArrayList

        ArrayList<Integer> a = new ArrayList<Integer>();

        // Adding Element in ArrayList

        a.add(1);

        a.add(2);

        a.add(3);

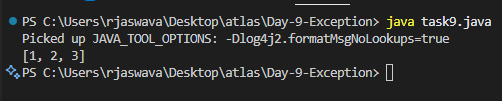
        // Printing ArrayList

        System.out.println(a);

    }

}

Output



Task 10

import java.util.\*;

public class task10 {

    public static void main(String args[]) {

        ArrayList<String> al = new ArrayList<>();

        al.add("Prasunamba");

        al.add("Meher");

        System.out.println("Original List : " + al);

        al.add(1, "Hello");

        System.out.println("After Adding element at index 1 : " + al);

        al.remove(0);

        System.out.println("Element removed from index 0 : " + al);

        al.remove("Prasunamba");

        System.out.println("Element Prasunamba removed : " + al);

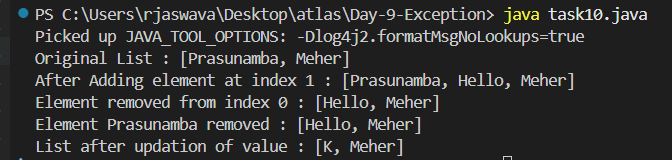
        al.set(0, "K");

        System.out.println("List after updation of value : " + al);

    }

}

Output



Task 11

class OuterClass {

    int x = 10;

    class InnerClass {

        int y = 5;

    }

}

public class task11 {

    public static void main(String[] args) {

        OuterClass myOuter = new OuterClass();

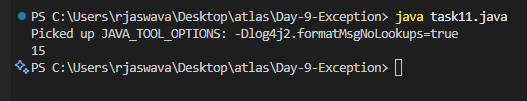
        OuterClass.InnerClass myInner = myOuter.new InnerClass();

        System.out.println(myInner.y + myOuter.x);

    }

}

Output



Task 12

class OuterClass {

    int x = 10;

    private class InnerClass {

        int y = 5;

    }

    // Method to access private inner class

    public int getSum() {

        InnerClass inner = new InnerClass();

        return this.x + inner.y;

    }

}

public class task12 {

    public static void main(String[] args) {

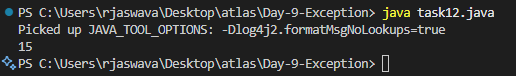
        OuterClass myOuter = new OuterClass();

        System.out.println(myOuter.getSum());

    }

}

Output



Task 13

class OuterClass {

    int x = 10;

    static class InnerClass {

        int y = 5;

    }

}

public class task13 {

    public static void main(String[] args) {

        OuterClass myOuter = new OuterClass();

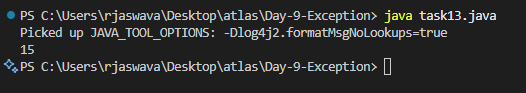
        OuterClass.InnerClass myInner = new OuterClass.InnerClass();

        System.out.println(myInner.y + myOuter.x);

    }

}

Output



Task 14

class OuterClass {

    int x = 50;

    class InnerClass {

        public int innerMethod() {

            return x;

        }

    }

}

public class task14 {

    public static void main(String[] args) {

        OuterClass myOuter = new OuterClass();

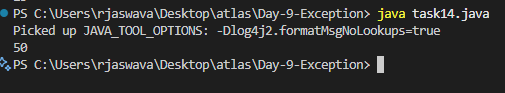
        OuterClass.InnerClass myInner = myOuter.new InnerClass();

        System.out.println(myInner.innerMethod());

    }

}

Output



Task 15

class OuterClass {

    int x = 10;

    static class InnerClass {

        static int y = 5;

    }

}

public class task15 {

    public static void main(String[] args) {

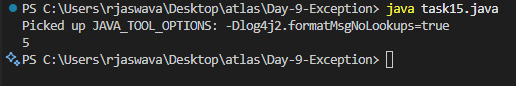
        OuterClass.InnerClass myInner = new OuterClass.InnerClass();

        System.out.println(myInner.y);

    }

}

Output



Task 16

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) {

        this.age = age;

    }

    @Override

    public String toString() {

        return "Name: " + name + ", Age: " + age;

    }

}

class Employee extends Person {

    private String employeeId;

    private double salary;

    public Employee(String name, int age, String employeeId, double salary) {

        super(name, age);

        this.employeeId = employeeId;

        this.salary = salary;

    }

    public String getEmployeeId() {

        return employeeId;

    }

    public void setEmployeeId(String employeeId) {

        this.employeeId = employeeId;

    }

    public double getSalary() {

        return salary;

    }

    public void setSalary(double salary) {

        this.salary = salary;

    }

    @Override

    public String toString() {

        return super.toString() + ", Employee ID: " + employeeId + ", Salary: " + salary;

    }

}

class Manager extends Employee {

    private String department;

    private int teamSize;

    public Manager(String name, int age, String employeeId, double salary, String department, int teamSize) {

        super(name, age, employeeId, salary);

        this.department = department;

        this.teamSize = teamSize;

    }

    public String getDepartment() {

        return department;

    }

    public void setDepartment(String department) {

        this.department = department;

    }

    public int getTeamSize() {

        return teamSize;

    }

    public void setTeamSize(int teamSize) {

        this.teamSize = teamSize;

    }

    @Override

    public String toString() {

        return super.toString() + ", Department: " + department + ", Team Size: " + teamSize;

    }

}

public class task16 {

    public static void main(String[] args) {

        Manager manager = new Manager("Mike Johnson", 40, "M301", 80000.0, "IT", 10);

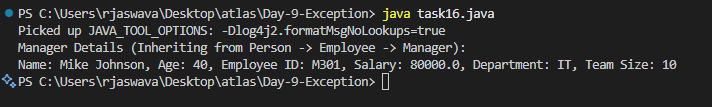
        System.out.println("Manager Details (Inheriting from Person -> Employee -> Manager):");

        System.out.println(manager);

    }

}

Output



Task 17:

What are the features of Java 8?

Java 8 Features:

Lambda Exp..

Interfaces can have body (default and static)

Foreach

Collection Api

Stream Api

Java io improvements

Functional Interfaces

Method references

Time ApI

Task 18

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.function.Consumer;

import java.lang.Integer;

class MyConsumer implements Consumer<Integer> {

    public void accept(Integer t) {

        System.out.println("Consumer impl Value::" + t);

    }

}

public class task18 {

    public static void main(String[] args) {

        List<Integer> myList = new ArrayList<Integer>();

        for(int i=0; i<10; i++) myList.add(i);

        // Using Iterator

        Iterator<Integer> it = myList.iterator();

        while(it.hasNext()) {

            Integer i = it.next();

            System.out.println("Iterator Value::" + i);

        }

        // Using forEach with anonymous class

        myList.forEach(new Consumer<Integer>() {

            public void accept(Integer t) {

                System.out.println("forEach anonymous class Value::" + t);

            }

        });

        // Using forEach with Consumer implementation

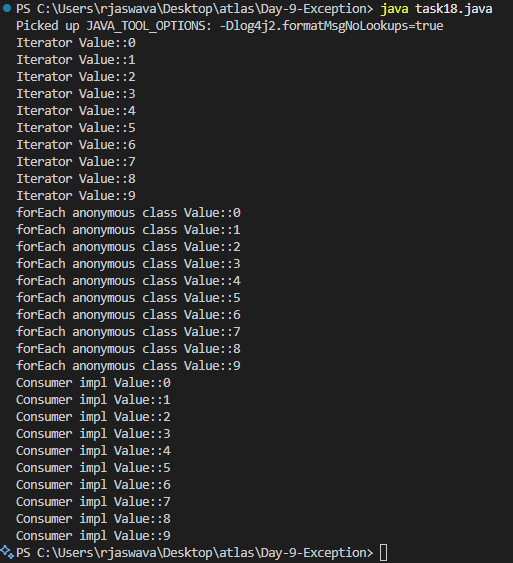
        MyConsumer action = new MyConsumer();

        myList.forEach(action);

    }

}

Output:



Task 20

import java.util.\*;

public class task20 {

    public static void main(String[] args) {

        // ArrayList

        List<String> a1 = new ArrayList<>();

        a1.add("Zara");

        a1.add("Mahnaz");

        a1.add("Ayan");

        System.out.println(" ArrayList Elements");

        System.out.print("\t" + a1);

        // LinkedList

        List<String> l1 = new LinkedList<>();

        l1.add("Zara");

        l1.add("Mahnaz");

        l1.add("Ayan");

        System.out.println();

        System.out.println(" LinkedList Elements");

        System.out.print("\t" + l1);

        // HashSet

        Set<String> s1 = new HashSet<>();

        s1.add("Zara");

        s1.add("Mahnaz");

        s1.add("Ayan");

        System.out.println();

        System.out.println(" Set Elements");

        System.out.print("\t" + s1);

        // HashMap

        Map<String, String> m1 = new HashMap<>();

        m1.put("Zara", "8");

        m1.put("Mahnaz", "31");

        m1.put("Ayan", "12");

        m1.put("Daisy", "14");

        System.out.println();

        System.out.println(" Map Elements");

        System.out.print("\t" + m1);

    }

}

Output

