**Java Programming**

**TASK1:**

**public class TASK1**

**{**

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

**{**

**Int[ ] myNumbers = {1, 2, 3};**

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

**}**

**}**

**Output:**

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 3

at TASK1.main(TASK1.java:5)

**TASK2:**

**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:**

Something went wrong.

**TASK3:**

**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**

Something went wrong.

I'm from finally block.

**TASK4:**

**public class TASK4 {**

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

**try {**

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

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

**System.*out*.println(myNumbers[2]);**

**} catch (Exception e) {**

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

**} finally {**

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

**}**

**}**

**}**

**Output**

2

3

I'm from finally block.

**TASK8**

**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.getMessage());**

**}**

**System.*out*.println("Out of Block");**

**}**

**}**

**Output**

Exception thrown:/ by zero

Out of Block

**TASK6**

**public class TASK5\_1 {**

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

**{**

**try**

**{**

**int a[] = new int[2];**

**int b= 10;**

**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.getMessage());**

**}**

**System.*out*.println("Out of Block");**

**}**

**}**

**Output**

ArrayIndexOutOfBoundsException thrown:java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2

Out of Block

**TASK7**

**public class TASK7**

**{**

**public static void fun() throws IllegalAccessException**

**{**

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

**throw new IllegalAccessException("Demo Exception in fun method.");**

**}**

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

**{**

**try**

**{**

***fun*();**

**}**

**catch(IllegalAccessException e)**

**{**

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

**}**

**}**

**}**

**Output**

Inside fun()

Caught in Main.Demo Exception in fun method.

**TASK8**

**class MyException extends Exception { // exception is a predefined class – parent class for MyException**

**public MyException(String m) {**

**super(m); // calling the parent class constructor with parameters**

**}**

**}**

**// A Class that uses the above MyException**

**public class TASK8 {**

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

**try {**

**// Throw an object of user-defined exception**

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

**}**

**catch (MyException ex) {**

**System.*out*.println("Caught");**

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

**}**

**}**

**}**

**Output**

Caught

This is a custom exception

**TASK9**

**import java.util.ArrayList;**

**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**

[1, 2, 3]

**TASK10**

**import java.util.\*;**

**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**

Original List : [Prasunamba, Meher]

After Adding element at index 1 : [Prasunamba, Hello, Meher]

Element removed from index 0 : [Hello, Meher]

Element Prasunamba removed : [Hello, Meher]

List after updation of value : [K, Meher]

**TASK11**

**What is the output of the below code snippet.. Explain ..**

**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**

15

**TASK12**

**Use the above code and make the inner class as private and see the output..**

**Ex: private class InnerClass {**

**class OuterClass6 {**

**int x = 10;**

**private class InnerClass {**

**int y = 5;**

**}**

**}**

**public class TASK12 {**

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

**OuterClass6 myOuter = new OuterClass6();**

**OuterClass6.InnerClass myInner = myOuter.new InnerClass();**

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

**}**

**}**

**Output**

Error as Inner class is not visible, because it is private.

**TASK13**

**Use the above code Task 011 and make the inner class static … see the output and explain..**

**Ex: static class InnerClass {**

**class OuterClass1 {**

**int x = 10;**

**static class InnerClass {**

**int y = 5;**

**}**

**}**

**public class TASK13 {**

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

**OuterClass1 myOuter = new OuterClass1();**

**OuterClass1.InnerClass myInner = new OuterClass1.InnerClass();**

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

**}**

**}**

**Output**

**15**

**TASK14**

**Use the above code Task 11 and create a method in innerclass and return the outer class variable**

**class OuterClass3**

**{**

**int x = 50;**

**class InnerClass**

**{**

**public int innerMethod()**

**{**

**return x;**

**}**

**}**

**}**

**public class TASK14**

**{**

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

**{**

**OuterClass3 myOuter = new OuterClass3();**

**OuterClass3.InnerClass myInner = myOuter.new InnerClass();**

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

**}**

**}**

**Output**

50

**TASK15**

Give reason or the code..

**class OuterClass4 {**

**int x = 10;**

**static class InnerClass {**

**static int *y* = 5;**

**}**

**}**

**public class TASK15 {**

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

**OuterClass4.InnerClass myInner = new OuterClass4.InnerClass();**

**System.*out*.println(myInner.*y*);**

**(or)**

**System.*out*.println(OuterClass4.InnerClass.*y*);**

**}**

**}**

**Output**

5

**TASK16**

**class Person2 {**

**private String name;**

**private int age;**

**public Person2() {}**

**public Person2(String name, int age) {**

**this.name = name;**

**this.age = age;**

**}**

**// Getters and setters**

**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 "Person [Name=" + name + ", Age=" + age + "]";**

**}**

**}**

**//Employee.java**

**class Employee1 extends Person2 {**

**private String employeeId;**

**private String department;**

**public Employee1() {}**

**public Employee1(String name, int age, String employeeId, String department) {**

**super(name, age);**

**this.employeeId = employeeId;**

**this.department = department;**

**}**

**// Getters and setters**

**public String getEmployeeId() { return employeeId; }**

**public void setEmployeeId(String employeeId) { this.employeeId = employeeId; }**

**public String getDepartment() { return department; }**

**public void setDepartment(String department) { this.department = department; }**

**@Override**

**public String toString() {**

**return super.toString() + ", Employee [Employee ID=" + employeeId + ", Department=" + department + "]";**

**}**

**}**

**//Manager.java**

**class Manager extends Employee1 {**

**private int teamSize;**

**private String project;**

**public Manager() {}**

**public Manager(String name, int age, String employeeId, String department, int teamSize, String project) {**

**super(name, age, employeeId, department);**

**this.teamSize = teamSize;**

**this.project = project;**

**}**

**// Getters and setters**

**public int getTeamSize() { return teamSize; }**

**public void setTeamSize(int teamSize) { this.teamSize = teamSize; }**

**public String getProject() { return project; }**

**public void setProject(String project) { this.project = project; }**

**@Override**

**public String toString() {**

**return super.toString() + ", Manager [Team Size=" + teamSize + ", Project=" + project + "]";**

**}**

**}**

**public class TASK16 {**

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

**{**

**Manager manager = new Manager("Alice", 40, "E123", "IT", 10, "AI Development");**

**System.*out*.println(manager.toString());**

**}**

**}**

**Output**

Person [Name=Alice, Age=40], Employee [Employee ID=E123, Department=IT], Manager [Team Size=10, Project=AI Development]

**TASK17**

What are the features of Java 8?

1. **Lambda Expressions**
2. **Functional Interfaces**
3. **Streams API**
4. **Default and Static Methods in Interfaces**
5. **Method References**
6. **New Date and Time API (in java.time package)**
7. **Collectors (with Streams)**

**TASK18**

**Run the code and give reason for the output..**

**import java.util.ArrayList;**

**import java.util.Iterator;**

**import java.util.List;**

**import java.util.function.Consumer;**

**import java.lang.Integer;**

**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);**

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

**while(it.hasNext()){**

**Integer i = it.next();**

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

**}**

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

**public void accept(Integer t) {**

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

**}**

**});**

**//traversing with Consumer interface implementation**

**MyConsumer action = new MyConsumer();**

**myList.forEach(action);**

**}**

**}**

**//Consumer implementation that can be reused**

**class MyConsumer implements Consumer<Integer>{**

**public void accept(Integer t) {**

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

**}**

**}**

**Output**

Iterator Value::0

Iterator Value::1

Iterator Value::2

Iterator Value::3

Iterator Value::4

Iterator Value::5

Iterator Value::6

Iterator Value::7

Iterator Value::8

Iterator Value::9

forEach anonymous class Value::0

forEach anonymous class Value::1

forEach anonymous class Value::2

forEach anonymous class Value::3

forEach anonymous class Value::4

forEach anonymous class Value::5

forEach anonymous class Value::6

forEach anonymous class Value::7

forEach anonymous class Value::8

forEach anonymous class Value::9

Consumer impl Value::0

Consumer impl Value::1

Consumer impl Value::2

Consumer impl Value::3

Consumer impl Value::4

Consumer impl Value::5

Consumer impl Value::6

Consumer impl Value::7

Consumer impl Value::8

Consumer impl Value::9

**TASK19**

**Plz go through this link for Java 8 features..**

**TASK20**

**import java.util.\*;**

**public class TASK20 {**

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

**// ArrayList**

**List a1 = new ArrayList();**

**a1.add("Zara");**

**a1.add("Mahnaz");**

**a1.add("Ayan");**

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

**System.*out*.print("\t" + a1);**

**// LinkedList**

**List 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 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 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**

ArrayList Elements

[Zara, Mahnaz, Ayan]

LinkedList Elements

[Zara, Mahnaz, Ayan]

Set Elements

[Ayan, Zara, Mahnaz]

Map Elements

{Daisy=14, Ayan=12, Zara=8, Mahnaz=31}

**TASK21**