

# Advanced Programming Language

## Assignment – 5

Chinmay Dorge

LCS2020022

Q1) Write a package which contains a class called abstract class Citizen. Accept following details through parameterized constructor

- 1) Name
- 2) Aadhar no
- 3) Phone number
- 4) Age

Citizen class also contains an abstract method called verifyAge() which checks if the person is above 18 years of age or not. Create a Vaccine class which will use the Citizen package and extend it. It will implement verifyAge() method .If the person is not eligible for vaccination then throw an exception .

Code: Citizen.java

```
package citizen;
public abstract class Citizen {
    protected String name, aadharNo, phoneNo;
    protected int age;

    public Citizen(String name , String aadharNo , String phoneNo , int
age){
        this.name = name;
        this.aadharNo = aadharNo;
        this.phoneNo = phoneNo;
        this.age = age;
    }

    public abstract void verifyAge() throws Exception;
}
```

Vaccine.java

```
package citizen;
import citizen.Citizen;
public class Vaccine extends Citizen{
```

```

    public Vaccine(String name , String aadharNo , String phoneNo , int
age){
        super(name,aadharNo,phoneNo,age);
    }

    public void verifyAge() throws Exception {
        if(age < 18){
            throw new Exception("Cannot be vaccinated. Age should be above
18");
        }
        else{
            System.out.println(name + " can be vaccinated");
        }
    }
}

```

## Main.java

```

package citizen;
import java.util.Scanner;
public class Main {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Question 1");
        System.out.println("-----");
        System.out.println();
        System.out.print("Enter name: ");
        String n = sc.nextLine();
        System.out.print("Enter Aadhar Card number: ");
        String a = sc.next();
        System.out.print("Enter mobile number: ");
        String m = sc.next();
        System.out.print("Enter age: ");
        int age = sc.nextInt();

        Vaccine v = new Vaccine(n,a,m,age);
        try{
            v.verifyAge();
        }catch(Exception e){
            System.out.println(e);
        }
    }
}

```

Output:

```
PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q1> java citizen.Main
Question 1
-----

Enter name: Chinmay
Enter Aadhar Card number: 866544430909
Enter mobile number: 987654321
Enter age: 19
Chinmay can be vaccinated
PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q1> java citizen.Main
Question 1
-----

Enter name: Dhruv
Enter Aadhar Card number: 456712348765
Enter mobile number: 912345678
Enter age: 10
java.lang.Exception: Cannot be vaccinated. Age should be above 18
```

Q2) Create a package that consists of two public classes. First class contains method which takes a list as input and returns sum of them. Second class contains method which also takes a list as input and returns minimum number from that array. Now create a main class and import this package and both these classes. In this main class given a List of numbers find:  
 $ans = \text{sum}(\text{arr}) / \text{min}(\text{arr})$ , where  $\text{sum}(\text{arr})$  returns sum of arr  
 $\text{min}(\text{arr})$  returns min of arr.

Return ans.

Implement appropriate error handling mechanisms to handle any possible exception that you might encounter.

Code: sum.java

```
package Q2;
public class sum {
    public int getSum(int[] arr) {
        int sum = 0;
        for (int i : arr)
            sum += i;
        return sum;
    }
}
```

## min.java

```
package Q2;
public class min {
    public int getMin(int[] arr) throws Exception{
        if (arr.length == 0)
            throw new Exception("Array size 0 - cannot compute minimum ");
        int min = arr[0];
        for (int i : arr)
            if (min > i)
                min = i;
        return min;
    }
}
```

## Main.java

```
import Q2.sum;
import Q2.min;
import java.util.Scanner;
public class Main {

    static float func(int[] arr , int sum , int min) throws Exception{
        if(min == 0){
            throw new Exception("Cannot divide by 0");
        }
        float a = (float)sum / (float)min;
        return a;
    }

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);
        int[] a = new int[5];
        System.out.print("Enter 5 enumbers: ");
        for(int i=0 ; i<5; i++){
            a[i] = sc.nextInt();
        }
        min m1 = new min();
        sum s1 = new sum();

        try{
            float ans = func(a , s1.getSum(a) , m1.getMin(a));
            System.out.println("Min = " + m1.getMin(a));
            System.out.println("Sum = " + s1.getSum(a));
            System.out.println("Ans = " + ans);
        }catch (Exception e){
            System.out.println(e);
        }
    }
}
```

Output:

```
PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q2> java Main
Enter 5 enumbers: 1 4 5 7 8
Min = 1
Sum = 25
Ans = 25.0
PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q2> java Main
Enter 5 enumbers: 0 0 0 0 0
java.lang.Exception: Cannot divide by 0
```

Q3) Write a java program that demonstrates checked and unchecked exceptions. Implement two methods, where first method throws 2 unchecked exceptions and second method throws 3 checked exceptions. Also handle these exceptions using appropriate try-catch and finally block.

Code: Main.java

```
import java.io.IOException;
import java.sql.SQLException;
import java.util.Scanner;

public class Main
{
    public void uncheckedExceptions()
    {
        System.out.print("Enter size of array: ");
        Scanner sc = new Scanner(System.in);
        int size = sc.nextInt();
        if(size<0)
            throw new IllegalArgumentException("Size cannot be negative");
        int[] arr = new int[size];
        for(int i=0;i<size;i++)
        {
            arr[i]=i;
        }
        System.out.print("Enter element you want to access: ");
        int pos = sc.nextInt();
        if(pos>size)
            throw new ArrayIndexOutOfBoundsException("Enter value between 0 and size-1");
        System.out.println("Element at pos "+pos+" = "+arr[pos-1]);

        System.out.print("Enter numerator: ");
        int num = sc.nextInt();
        System.out.print("Enter denominator: ");
        int den = sc.nextInt();
        if(den==0)
            throw new ArithmeticException("Denominator cannot be zero");
        System.out.println("Ans = "+num/den);
    }
}
```

```

    public void checkedException(int n) throws
IOException, SQLException, ClassNotFoundException
    {
        if(n==1)
            throw new IOException("IO exception");
        if(n==2)
            throw new SQLException("Sql exception");
        if(n==3)
            throw new ClassNotFoundException("Class not found exception");
    }

    public static void main(String[] args){
        Main obj = new Main();
        try {
            obj.uncheckedExceptions();
        } catch (IllegalArgumentException e) {
            System.out.println(e);
        } catch (ArrayIndexOutOfBoundsException g) {
            System.out.println(g);
        } catch (ArithmeticException f) {
            System.out.println(f);
        }

        System.out.print("Enter number: ");
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        try
        {
            obj.checkedException(num);
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        catch(SQLException f)
        {
            f.printStackTrace();
        }
        catch(ClassNotFoundException c)
        {
            c.printStackTrace();
        }
    }
}

```

Output:

```

PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q3> java Main
Enter size of array: 4
Enter element you want to access: 5
java.lang.ArrayIndexOutOfBoundsException: Enter value between 0 and size-1
Enter number: 2
java.sql.SQLException: Sql exception
    at Main.checkedException(Main.java:40)
    at Main.main(Main.java:62)

```

```

PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q3> java Main
Enter size of array: 5
Enter element you want to access: 4
Element at pos 4 = 3
Enter numerator: 9
Enter denominator: 0
java.lang.ArithmeticException: Denominator cannot be zero
Enter number: 1
java.io.IOException: IO exception
    at Main.checkedException(Main.java:38)
    at Main.main(Main.java:62)

```

```

PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q3> java Main
Enter size of array: 6
Enter element you want to access: 4
Element at pos 4 = 3
Enter numerator: 3
Enter denominator: 1
Ans = 3
Enter number: 3
java.lang.ClassNotFoundException: Class not found exception
    at Main.checkedException(Main.java:42)
    at Main.main(Main.java:62)

```

Q4) Write a java program which contains 4 methods says first, second, third and fourth. Main method will call first, first will call second, second will call third and third will call fourth. In fourth method throw an ArithmeticException and propagate this exception up to the main method and handle it in main method.

Code: Q4.java

```

public class Q4 {
    static void first() {
        second();
        System.out.println("Handled in first");
    }
    static void second() {
        third();
        System.out.println("Handled in second");
    }
    static void third() {
        fourth();
        System.out.println("Handled in third");
    }
}

```

```

static void fourth() throws ArithmeticException{
    throw new ArithmeticException("Exception created in fourth");
}

public static void main(String[] args){
    try{
        first();
    }
    catch (Exception e){
        System.out.println(e);
        System.out.println("Handled in Main");
    }
}
}

```

Output:

```

PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q4> java Q4
java.lang.ArithmeticException: Exception created in fourth
Handled in Main

```

Q5) Finally will always be executed whether exception is handled or not. Even if you return from the method in the try block, finally block will still be executed. Write a java program which demonstrates this concept. In finally block throw an exception, which is different from the exception thrown in the try block. Handle these exceptions appropriately.

Code: withoutException.java

```

public class withoutException {
    public static void main(String[] args) throws ArithmeticException{
        try {
            System.out.println("Try Block");
            int a = 50 / 10;
        } catch (Exception e){
            throw new ArithmeticException("Cannot divide by 0");
        } finally {
            System.out.println("Finally statement");
        }
        System.out.println("Rest of code");
    }
}

```

withException.java

```

public class withException {
    public static void main(String[] args) throws ArithmeticException{
        try {
            System.out.println("Try block");
            int a = 50 / 0;
        } catch (Exception e){
            throw new ArithmeticException("Cannot divide by 0");
        }
    }
}

```



```
    }finally {  
        System.out.println("Finally statement");  
    }  
    System.out.println("Rest of code");  
}  
}
```

Output:

```
PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q5> java withoutException  
Try Block  
Finally statement  
Rest of code  
PS C:\Users\hp\Desktop\CLG\Third Sem\APL\Assignment-5\Q5> java withException  
Try block  
Finally statement  
Exception in thread "main" java.lang.ArithmeticException: Cannot divide by 0  
    at withException.main(withException.java:7)
```

Note: The code has been sent with the zip file and is also available on GitHub. Repo link

<https://github.com/Chinmay-Dorge/Advanced-Programming-Assignments>