FOCP II

Lab Manual

Department of Computer Science and Engineering
The NorthCap University, Gurugram

FOCP II

Lab Manual

CSL 108

Dr. Neeti Kashyap

Department of Computer Science and Engineering

NorthCap University, Gurugram- 122001, India

Session 2021-22

Published by:

School of Engineering and Technology

Department of Computer Science & Engineering

The NorthCap University Gurugram

• Laboratory Manual is for Internal Circulation only

Copying or facilitating copying of lab work comes under cheating and is considered as use of unfair means. Students indulging in copying or facilitating copying shall be awarded zero marks for that particular experiment. Frequent cases of copying may lead to disciplinary action. Attendance in lab classes is mandatory.

Labs are open up to 7 PM upon request. Students are encouraged to make full use of labs beyond normal lab hours.

PREFACE

The aim of this lab manual is to help students understand real life problem using programming skills.

This manual is required for second semester computer science and engineering students, so that they are able understand Java, one of the most in-demand programming languages. The lab manual outline is designed in such a manner that the beginners with little or no knowledge about Object Oriented programming concepts can understand the core OOP concepts including Encapsulation, Polymorphism, Inheritance etc. and their implementation in Java. The students will have extensive hands-on experience writing, compiling, testing and executing Java programs applying the above principles for developing modular reusable programs. to understand fundamental concepts of Java programming language that can be further used to design applications.

By the end of this practical, the students will gain the foundational skills a software engineer needs, to solve real-world problems, from designing algorithms to testing and debugging; and will be able to apply these concepts to build their own interactive Java applications.

Author expresses deep gratitude to Members, Governing Body-NCU for encouragement and motivation.

Dr. Neeti Kashyap
The NorthCap University
Gurugram, India

CONTENTS

1. INTRODUCTION

That 'learning is a continuous process' cannot be over emphasized. The theoretical knowledge gained during lecture sessions need to be strengthened through practical experimentation. Thus, practical makes an integral part of a learning process.

The purpose of conducting experiments can be stated as follows:

- To familiarize the students with the basic concepts, programming skill development and the take home laboratory assignments mainly implementation-oriented which have to be coded in high level language. The lab sessions will be based on exploring the concepts discussed in class.
- Observing basic structure and characteristics of Computer Systems
- Reporting and analyzing the programming concepts.
- Hands on experience on the coding and software tools

2. LAB REQUIREMENTS

- Mac or PC
- Java Development kit (JDK)
- Free Text editor (Sublime Recommended)
- Web Browser (Chrome and Firefox Recommended)
- GIT user Account

3. GENERAL INSTRUCTIONS

3.1 General discipline in the lab

- Students must turn up in time and contact concerned faculty for the experiment they are supposed to perform.
- Students will not be allowed to enter late in the lab.
- Students will not leave the class till the period is over.
- Students should come prepared for their experiment.
- Experimental results should be entered in the lab report format and certified/signed by concerned faculty/ lab Instructor.

- Students must get the connection of the hardware setup verified before switching on the power supply.
- Students should maintain silence while performing the experiments. If any necessity arises for discussion amongst them, they should discuss with a very low pitch without disturbing the adjacent groups.
- Violating the above code of conduct may attract disciplinary action.
- Damaging lab equipment or removing any component from the lab may invite penalties and strict disciplinary action.

3.2 Attendance

- Attendance in the lab class is compulsory.
- Students should not attend a different lab group/section other than the one assigned at the beginning of the session.
- On account of illness or some family problems, if a student misses his/her lab classes, he/she may be assigned a different group to make up the losses in consultation with the concerned faculty / lab instructor. Or he/she may work in the lab during spare/extra hours to complete the experiment. No attendance will be granted for such case**.**

3.3 Preparation and Performance

- Students should come to the lab thoroughly prepared on the practicals they are assigned to perform on that day. Brief introduction to each experiment with information about self-study reference is provided on LMS.
- Students must bring the lab report during each practical class with written records of the last experiments performed complete in all respect.
- Each student is required to write a complete report of the practical he has performed and bring to lab class for evaluation in the next working lab. Sufficient space in work book is provided for independent writing of theory, observation, calculation and conclusion.
- Students should follow the Zero tolerance policy for copying / plagiarism. Zero marks will be awarded if found copied. If caught further, it will lead to disciplinary action.
- Refer **Annexure 1** for Lab Report Format

3.4 Norms to be followed for doing lab practical

- All the students will work in a team of 2 members each.
- Each and every practical will contain a number of programming problems to be solved by students.

4. LIST OF EXPERIMENTS

Exp. No.	List of Experiments
1	Programs on Data types
2	Programs on Control Statements
3	Programs on Arrays
4	Programs on classes and objects
5	Programs on Inheritance
6	Programs on Interface
7	Programs on Packages
8	Programs on Exception Handling
9	Programs on File Handling

5. RUBRICS

Marks Distribution	
Continuous Evaluation (50 Marks)	**End Semester Project (20 Marks)**
Each experiment shall be evaluated for 10 marks and at the end of the semester proportional marks shall be awarded out of 50.	Unguided project carries 20 marks.

6. Annexure 1

FOCP II

CSL108

Project Report

Faculty name:

Student name:

Roll No.:

Semester:

Group:

Department of Computer Science and Engineering

The NorthCap University, Gurugram- 122001, India

Session 2021-22

Table of Contents

Index

S No	Experiment	Date of Experiment	Date of Submission	Marks	CO Covered	Signature
1	Programs on Data Types					
2	Programs on Control Statements					
3	Programs on Arrays					
4	Programs on Classes & Objects					
5	Programs on Inheritance					
6	Programs on Packages					
7	Programs on Interfaces					
8	Programs on Exception Handling					
9	Programs on File handling					

PRACTICAL NO. 1

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective(s):

- Perform variable assignment.
- Use comments in code
- Use operator precedence and operator associativity.
- Effectively use arithmetic expressions in Java

Outcome:

Student will be familiarizing with the data types in Java.

Problem Statement:

Q1. Write a program that takes two integers (values to be given within the program) and displays the output of the following operators: addition, subtraction, multiplication, division and modulus.
Definition of Done:
DoD 1: Assign two numbers to two variables.
DoD2: Use separate variables to store the results.

Q2. Write a Java program to convert minutes into the number of years, months and days.
Definition Of Done :
DoD 1: Ask the user to input the minutes
DoD 2: Display years and months and days in the sequence yy-mm-dd.

Background Study: Data Types in Java

Primitive data types: The primitive data types include boolean, char, byte, short, int, long, float and double.
Non-primitive data types: The non-primitive data types include Classes, Interfaces, and Arrays.

Question Bank:

1. Why Java is considered dynamic?

2. What is Java Virtual Machine and how it is considered in context of Java's platform independent feature?

3. List two Java IDE's? List some Java keywords(unlike C, C++ keywords)?

4. Consider the following class:

```
public class IdentifyMyParts {
public static int x = 7;
public int y = 3;
}
```

a) What are the class variables?

b) What are the instance variables?

Flipped practicals

1. What is the output from the following code:

```
IdentifyMyParts a = new IdentifyMyParts();
IdentifyMyParts b = new IdentifyMyParts();
a.y = 5;
b.y = 6;
a.x = 1;
b.x = 2;
System.out.println("a.y = " + a.y);
System.out.println("b.y = " + b.y);
System.out.println("a.x = " + a.x);
System.out.println("b.x = " + b.x);
System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);
```

2. What's wrong with the following program?

```
public class SomethingIsWrong {
    public static void main(String[] args) {
        Rectangle myRect;
        myRect.width = 40;
        myRect.height = 50;
        System.out.println("myRect's area is " + myRect.area());
    }
}
```

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO. 2

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective

To familiarize the students with control statements in JAVA

Program Outcome

Through this practical, students will learn about the control statements

Problem Statement

1. Write a Java program to solve quadratic equations (use if, else if and else).

Definition of Done:
DoD 1: The program asks the values of coefficients of a quadratic equation.
DoD 2: The program should display the roots or an appropriate message.

2. Write a Java program that asks the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.

Definition of Done:
DoD 1: The program asks an input from the user.
DoD 2: A single character input is taken from the user or an error message is generated.
DoD 3: The program should print “Vowel” if the entered character is a vowel and “Consonant” if the entered character is a consonant.

3. Write a Java program to print following structure:

```
*
**
***
****
*****
******
```

Background Study:

Java compiler executes the java code from top to bottom. The statements are executed according to the order in which they appear. However, Java provides statements that can be used to control the flow of java code. Such statements are called control flow statements.

Java provides three types of control flow statements.

- Decision Making statements
- Loop statements
- Jump statements

Question Bank

1. The most basic control flow statement supported by the Java programming language is the ___ statement.
2. The ___ statement allows for any number of possible execution paths.
3. The ___ statement is similar to the while statement, but evaluates its expression at the ___ of the loop.
4. How do you write an infinite loop using the for statement?
5. How do you write an infinite loop using the while statement?
6. Which looping process checks the test condition at the end of the loop?
7. Why do we use continue statement?
8. What is the size of boolean variable?
9. Which looping process is best used when the number of iterations is known?

Flipped Practicals

1. Consider the following code snippet.

```
if (aNumber >= 0)
if (aNumber == 0)
System.out.println("first string");
else System.out.println("second string");
System.out.println("third string");
```

a) What output do you think the code will produce if aNumber is 3?
b) Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?
c) Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.
d) Use braces, { and }, to further clarify the code.

2. What's wrong? for (int k = 2, k <= 12, k++)
3. If there is more than one statement in the block of a for loop, what must be placed at the beginning and the ending of the loop block?

4. What value is stored in num at the end of this looping?

```
for (num = 1; num <= 5; num++)
```

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO. 3

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective To familiarize the students with array in JAVA.
Program Outcome The students will learn the concept of arrays in Java.
Problem Statement 1. Write a Java program to find the maximum and minimum value of an array. Definition of Done: DoD 1: The program should ask the user to enter the elements of the array. DoD 2: The program should display the maximum and minimum elements of the array. 2. Write a Java program to find the index of an array element in an array of size 10. The program should not use any function other than main ( ) functions. Definition of Done: DoD 1: The program should ask the user to enter the elements of the array. DoD 2: The program should ask the user to enter a number to search. DoD 3: The program should display the elements of the array entered. DoD 4: The program should display the index of the number if the item is present or display -1 of the elements is not present.

3. Write a Java Program to count even and odd numbers in an array.

Definition of Done

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: Even elements will be stored in EvenArray[] and odd elements will be stored in oddArray[].

DoD 3: Display all three arrays along with their length.

4. Write a Java program to read numbers in an integer array of size 5 and display the following (using functions for each functionality):

i) Sum of all the elements

ii) Sum of alternate elements in the array.

Definition of Done

DoD 1: The program should ask the user to enter the elements of the array.

DoD 2: The program should display a menu with the above choices and ask the user to choose one of the choices.

Background Study:

Java array is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

Question Bank:

1. Can you pass the negative number as an array size?

2. Can you change the size of the array once you define it?

3. What is an anonymous array?

4. What is the difference between int[] a and int a[] ?

5. What are jagged arrays in java? Give example?

Flipped Practicals

1.Which of these is an incorrect array declaration?
a) int arr[] = new int[5]
b) int [] arr = new int[5]
c) int arr[] = new int[5]
d) int arr[] = int [5] new

2. What will be the output of the following program?

```
public class MyFirst {
      public static void main(String[] args) {
         MyFirst obj = new MyFirst(n);
 }
 static int a = 10;
 static int n;
 int b = 5;
 int c;
 public MyFirst(int m) {
       System.out.println(a + ", " + b + ", " + c + ", " + n + ", " + m);
   }
// Instance Block
  {
     b = 30;
     n = 20;
  }
```

```
// Static Block
  static
{
        a = 60;
    }
 }
```

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO: 4

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective To familiarize the students with classes and objects.
Program Outcome The students will learn the concept of classes and objects.
Problem Statement 1.Create a class named 'Student' with String variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as '2' and that of name as "John" by creating an object of the class Student. 2. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor. 3. Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of the rectangle are entered through the keyboard. 4. Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by the user.

5. Write a program to calculate the distance between two points (x1, y1) and (x2, y2). All numbers and return values should be of type double.

Definition of Done:

DoD 1: Two java files to be defined. One for class definitions and another for the application

DoD 2: A class point is defined with two float variables for x1 and x2 and the following functionality:

i. Non-parametrized and parameterized constructors are defined.

ii. Get and set methods are defined for all the instance variables.

iii. Distance function is defined to calculate the distance between two points.

iv. Display function is defined with width of 7 and precision of 2.

Write this program with a static method definition for calculating the distance between two points.

Background Study

In object-oriented programming technique, we design a program using objects and classes. An object in Java is the physical as well as a logical entity, whereas, a class in Java is a logical entity only.

An object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

Question Bank

1. What is the difference between class and object?
2. What is constructor chaining?
3. What is No-arg constructor?
4. What happens if you keep return type for a constructor?

5. What is the use of private constructor?
6. Can we use this() in a method?
7. Can we define a method with same name of class?

Flipped Practicals

1. What will be the output of the following Java program?

```
class A
{
        int i;
        int j;
    A()
    {
        i = 1;
        j = 2;
    }
}
class Output
{
    public static void main(String args[])
    {
        A obj1 = new A();
        A obj2 = new A();
            System.out.print(obj1.equals(obj2));
    }
}
```

a) false
b) true
c) 1
d) Compilation Error

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO: 5

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective
To familiarize the students with the concept of Inheritance.

Program Outcome
The students will learn the concept of inheritance.

Program Statement

1. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

2. Create three classes:

Class Vehicle:

Vehicle Class will contain a display() function, which will say "This is a Vehicle".

Class Car:
Car Class will derive the Vehicle Class and overwrite its display() function. it will say "This is a Car".

Class Bike:

Bike Class will derive the Vehicle Class and overwrite its display() function. it will say "This is a Bike".

Write an application that reads an Integer N, which will denote the number of tyres in the vehicle. You have to create an object of the appropriate class according to the value of N and use it display() function.

If N = 2, Create a Bike Object.

If N = 4, Create a Car Object.

Create a Vehicle Object, otherwise.

Definition of Done:

DoD 1: Each class definition is stored in its own .java file.

DoD 2: Switch statement is used for identifying the appropriate class for which the object is to be invoked.

3. Define a class Box with the following instance variables: width, height and depth, all of type float. Create a new class BoxWeigth that extends Box to include weight as an instance variable. Write an application that tests the functionalities of both these classes.

Definition of Done:

DoD 1: Three java files to be defined. One for each class definition: Box, BoxWeight and BoxWeightDemo.
DoD 2: Box and BoxWeight should have three types of constructors defined: clone of an object, all dimensions specified as arguments, no argument.
DoD 3: Super is used to call base class constructors in derived class
DoD 4: Get and set functions defined as applicable in Box and BoxWeight classes.
DoD 5: Function to display volume in Box class and weight in BoxWeigth class

Background Study

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance, the information is made manageable in a hierarchical order.

The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

extends Keyword

extends is the keyword used to inherit the properties of a class. Following is the syntax of extends keyword.

Syntax

```
class Super {
    .....
    .....
}
class Sub extends Super {
    .....
    .....
}
```

Question Bank

1. What is the use of super keyword?
2. What is multi-level inheritance?
3. What is the usage of inheritance?

Flipped Questions

Q1. What is the output of the following?

```
class A
{
    {
        System.out.println(1);
    }
}

class B extends A
{
    {
        System.out.println(2);
    }
}
```

```
class C extends B
{
    {
        System.out.println(3);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

Q2. What is the output of the following?

```
class A
{
    public A()
    {
        System.out.println("Class A Constructor");
    }
}

class B extends A
{
    public B()
    {
        System.out.println("Class B Constructor");
    }
}

class C extends B
{
    public C()
    {
        System.out.println("Class C Constructor");
    }
}

public class MainClass
{
    public static void main(String[] args)
```

```
    {
        C c = new C();
    }
}
```

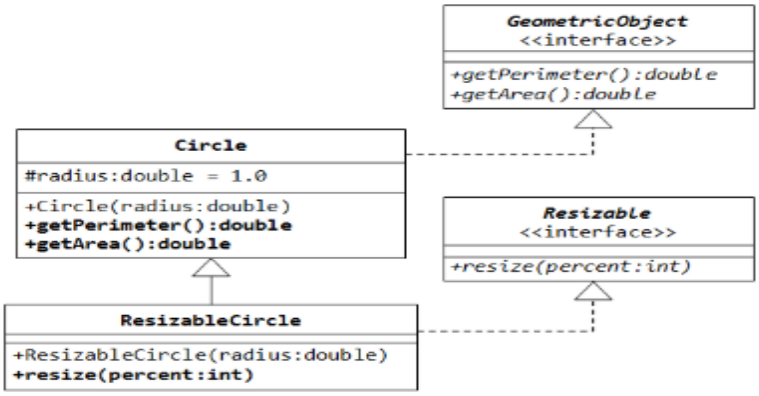
Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO. 6

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective To familiarize the students with the concept of Interfaces in Java.
Program Outcome The students will be able to understand where and how interfaces are implemented.
Problem Statement 1. a) Write a program in java to check if a class can extends another class and/ can implement one and more than one interface. b) Write a program in java to check if an interface can extend other interface. c) Write a program in java to check if an interface can also extend multiple interfaces. 2. Define the interface / class hierarchy as detailed in the following class diagram **Definition of Done:** DOD 1: The class definitions are defined as per the class diagram. DOD 2: Each class definition is stored in its own .java file. DOD 3: Base class constructors are invoked using super keyword DOD 4: Function overriding is applied wherever applicable.



3. We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

Background Study

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods. The interface in Java is *a mechanism to achieve* abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.
In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

Flipped Practicals

1. What is the output of this program?

```
interface calculate {
   void cal(int item);
}
class display implements calculate {
```

```
    int x;
    public void cal(int item) {
        x = item * item;
    }
}
class interfaces {
    public static void main(String args[]) {
        display arr = new display;
        arr.x = 0;
        arr.cal(2);
        System.out.print(arr.x);
    }
}
```

a) 0

b) 2

c) 4

d) None of the mentioned

2. Which of the following package stores all the standard java classes?

a) lang

b) java

c) util

d) java.packages

3. Determine output of the following code.

```
interface A { }
class C { }
class D extends C { }
```

```
class B extends D implements A { }
public class Test extends Thread{
    public static void main(String[] args){
        B b = new B();
        if (b instanceof A)
            System.out.println("b is an instance of A");
        if (b instanceof C)
            System.out.println("b is an instance of C");
    }
}
```

a) b is an instance of A.

b) b is an instance of C.

c) b is an instance of A followed by b is an instance of C.

Question Bank

1. Can an interface be final?
2. Can an abstract class implement an interface?
3. Can you declare an interface method static?
4. What is the difference between abstract class and interface?

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Experiment No: 7

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective
To familiarize the students with package in Java.

Program Outcome
The students will learn the concept of package in Java. They will be able to understand

Problem Statement

1. Create a Java package called exercises. Inside the exercises package, create another package (subpackage) called java. Create a Java class called PackageDemo inside the java package. Insert a display() method inside the PackageDemo class. Inside the method, insert this statement:System.out.println("PackageDemo executed");Write a PackageDemoDriver class within the same package to run the display () method of PackageDemo class.

2. Write a java program outside the above defined package that imports the package PackageDemo and calls its display() function.

Background Study

A **java package** is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Flipped Practicals

1. What will be the output of the following Java program?

```
package pkg;
class display
{
    int x;
    void show()
    {
        if (x > 1)
            System.out.print(x + " ");
    }
}
class packages
{
    public static void main(String args[])
    {
        display[] arr=new display[3];
        for(int i=0;i<3;i++)
            arr[i]=new display();
        arr[0].x = 0;
        arr[1].x = 1;
        arr[2].x = 2;
        for (int i = 0; i < 3; ++i)
            arr[i].show();
     }
}
```

Note : packages.class file is in directory pkg;

a) 0
b) 1
c) 2
d) 0 1 2

Question Bank

1. What are packages? what is use of packages ?

2. What is difference between importing "java.applet.Applet" and "java.applet.* "?

3.What do you understand by package access specifier?

4. By default,all program import the java.lang package.
True/False ?

5. Java compiler stores the .class files in the path specified in CLASSPATH environmental variable. True/False ?

6. User-defined package can also be imported just like the standard packages
True/False ?

7. A _____ is used to separate the hierarchy of the class while declaring an Import statement.

8. All standard classes of Java are included within a package called _____.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO.8

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective
To familiarize the students with exception in java.

Program Outcome
The students will learn the concept of exceptions in Java.

Problem Statement

1. Write a program that takes 5 integer command line arguments. Create a user defined Exception named CheckArgumentException to check the number of arguments passed through command line. If the number of arguments is less than five, throw the CheckArgumentException, else print the addition of all five numbers.

   Definition of Done:

   DOD 1:Create a user-defined exception by the name CheckArgumentException
   DOD 2:Ask the user to enter the number of arguments
   DOD 3:Use for loop to enter the arguments
   DOD 4: Calculate the sum of the values entered

2. Create a class with a main() method that *throws* an object of class **Exception** inside a *try* block. Give the constructor for **Exception** a String argument. Catch the exception inside a *catch* clause and print the String

argument. Add a *finally* clause and print a message to prove you were there.

Background Study

An exception (or exceptional event) is a problem that arises during the execution of a program. When an **Exception** occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.

An exception can occur for many different reasons. Following are some scenarios where an exception occurs.

- A user has entered an invalid data.
- A file that needs to be opened cannot be found.
- A network connection has been lost in the middle of communications or the JVM has run out of memory.

Question Bank

1. When does Exceptions in Java arises in code sequence?
a) Run Time
b) Compilation Time
c) Can Occur Any Time
d) None of the mentioned

3. Which of these keywords must be used to monitor for exceptions?
a) try
b) finally
c) throw
d) catch

4. Which of these keywords must be used to handle the exception thrown by try block in some rational manner?
a) try
b) finally

c) throw
d) catch

5. Which of these keywords is used to manually throw an exception?
a) try
b) finally
c) throw
d) catch

Flipped Practicals

1. What will be the output of the following Java program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
      {
         System.out.print("Hello" + " " + 1 / 0);
      }
      catch(ArithmeticException e)
      {
            System.out.print("World");
      }
   }
}
```

a) Hello
b) World
c) HelloWorld
d) Hello World

2. What will be the output of the following Java program?

```
class exception_handling
{
   public static void main(String args[])
   {
      try
     {
```

```
            int a, b;
            b = 0;
            a = 5 / b;
            System.out.print("A");
        }
        catch(ArithmeticException e)
        {
                System.out.print("B");
        }
    }
}
```

a) A
b) B
c) Compilation Error
d) Runtime Error

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

PRACTICAL NO. 9

Student Name and Roll Number:
Semester /Section:
Date:
Faculty Signature:

Objective
To familiarize the students with File handling.

Program Outcome
The students will learn the concept of files in Java.

Problem Statement

1. Write a program to check if the file exist is a file or directory.

2. Write a program to change the file permissions.

3. Write a program to perform simple read and write operation into file.

4. Write a program that writes an Serial No.(int), First Name (String), CGPA (float) and Grade(char) into a text file using bufferedWriter and displays the contents using bufferedReader.

Background Study

File handling is an important part of any application. Java has several methods for creating, reading, updating, and deleting files.

Java File Handling

The File class from the java.io package, allows us to work with files.

To use the File class, create an object of the class, and specify the filename or directory name:

Example

```
import java.io.File;  // Import the File class


File myObj = new File("filename.txt"); // Specify the filename
```

Question Bank

1. Which of these exception is thrown in cases when the file specified for writing is not found?
a) IOException
b) FileException
c) FileNotFoundException
d) FileInputException

2. Which of these methods are used to read in from file?
a) get()
b) read()
c) scan()
d) readFileInput()

3. Which of these values is returned by read() method is end of file (EOF) is encountered?
a) 0
b) 1
c) -1
d) Null

Flipped Practicals

1. What will be the output of the following Java program?

```
1.  import java.io.*;
2.  class filesinputoutput
3.  {
4.     public static void main(String args[])
5.     {
6.        InputStream obj = new FileInputStream("inputoutput.java");
7.        System.out.print(obj.available());
8.     }
9.  }
```

Note: inputoutput.java is stored in the disk.
a) true
b) false
c) prints number of bytes in file
d) prints number of characters in the file

2. What will be the output of the following Java program?

```
1.  import java.io.*;
2.  public class filesinputoutput
3.  {
4.            public static void main(String[] args)
5.     {
6.   String obj  = "abc";
7.        byte b[] = obj.getBytes();
8.        ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
9.        for (int i = 0; i < 2; ++ i)
10.        {
11.           int c;
12.           while((c = obj1.read()) != -1)
13.           {
14.             if(i == 0)
15.              {
16.                System.out.print(Character.toUpperCase((char)c));
17.                 obj2.write(1);
18.             }
19.           }
20.           System.out.print(obj2);
```

```
        }
    }
}
```

a) AaBaCa
b) ABCaaa
c) AaaBaaCaa
d) AaBaaCaaa

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs