

SHREE RAMCHANDRA EDUCATION SOCIETY'S
**SHREE RAMCHANDRA COLLEGE OF ENGINEERING,
LONIKAND, PUNE – 412 216**

DEPARTMENT OF ELECTRICAL ENGINEERING



LAB MANUAL

FUNDAMENTALS OF MICROPROCESSOR AND MICROCONTROLLER (FMM)
SE (EE) Semester-II

Prepared by
Prof. Ms. Arbale S.P. (Assistant Professor)


List of Experiments (S.P.UNIPUNE SYLLABUS)

Note: - Experiment-1 & experiment-8 is compulsory; select any 4 experiments from experiment-2 to experiment-7 and any 4 Experiments from experiment-9 to experiment-14.

Sr. No.	Name of Experiment
1	Assembly language Programming using 8085 (4 programs minimum based on 8 bit addition, 16 bit addition, multiplication, largest number, smallest number, ascending order, descending order).
2	Interfacing of 8255 with 8085.
3	Interfacing of 8254 with 8085.
4	Interfacing of 8 bit D/A and A/D converter with 8085.
5	Control of D.C. motor using 8085.
6	Measurement of speed using 8085.
7	Interfacing of seven segment LED display with 8085.
8	Assembly language Programming using 8051 (4 programs minimum based on 8 bit addition of 10 numbers, multiplication, largest number, smallest number, Ascending order, Descending order).
9	Control of stepper motor using 8051.
10	Measurement of temperature using 8051.
11	Interfacing of ADC 0809 with 8051 microcontroller
12	Interfacing of DAC 0800 with 8051 microcontroller
13	Interfacing of LCD with 8051.
14	Interfacing of 4 X 4 matrix keyboard.

INDEX

Sr. No.	Name of Experiment	Page No.
1.	Assembly language Programming using 8085 (4 programs minimum based on 8 bit addition, 16 bit addition, multiplication, largest number, smallest number, ascending order, descending order)	
2.	Interfacing of 8255 with 8085.	
3.	Interfacing of 8254 with 8085.	
4.	Interfacing of 8 bit D/A and A/D converter with 8085.	
5.	Control of D.C. motor using 8085.	
6.	Assembly language Programming using 8051 (4 programs minimum based on 8 bit addition of 10 numbers, multiplication, largest number, smallest number, Ascending order, Descending order).	
7.	Interfacing of ADC 0809 with 8051 microcontroller	
8.	Interfacing of DAC 0800 with 8051 microcontroller	
9.	Interfacing of LCD with 8051.	
10.	Interfacing of 4 X 4 matrix keyboard.	

	SHREE RAMCHANDRA COLLEGE OF ENGG. LONIKAND	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: ASSEMBLY LANGUAGE PROGRAMMING USING 8085	
DEPARTMENT OF ELECTRICAL ENGINEERING		
EXPERIMENT NO. : SRCOE/ELECT/SE/FMM/	SEMESTER : II (SE)	PAGE:

EXPERIMENT NO. 1

Title: ASSEMBLY LANGUAGE PROGRAMMING USING 8085

Aim:

To find the smallest element in an array.

ALGORITHM:

1. Place all the elements of an array in the consecutive memory locations.
2. Fetch the first element from the memory location and load it in the accumulator.
3. Initialize a counter (register) with the total number of elements in an array.
4. Decrement the counter by 1.
5. Increment the memory pointer to point to the next element.
6. Compare the accumulator content with the memory content (next element).
7. If the accumulator content is smaller, then move the memory content (largest element) to the accumulator. Else continue.
8. Decrement the counter by 1.
9. Repeat steps 5 to 8 until the counter reaches zero
10. Store the result (accumulator content) in the specified memory location.

Software required:

Theory:

Algorithm:

Class diagram:

Result:

Conclusion:



Experiment No. 2

Title: Write a program in C++ to perform following operations on complex numbers Add, Subtract, Multiply, Divide, Complex conjugate. Design the class for complex number representation and the operations to be performed.

Aim: The objective of this assignment is to learn the concepts classes and objects.

Software required:

1. Turbo C++

Theory:

1. Classes
2. Objects.

Algorithm:

1. Create the class of type complex.
2. Declare the function as public.
 - a) Void add ();
 - b) Void subtract ();
 - c) Void multiply ();
 - d) Void divide ();
 - e) Complex conjugate();
3. Function definition add () is used to add two complex numbers.
4. Function definition subtract () is used to subtract two complex numbers.
5. Function definition multiply () is used to multiply two complex numbers.
6. Function definition divide () is used to divide two complex numbers.
7. Function definition complex () is used to find out the complex conjugate of two number.
8. In the main () create the objects for class complex to call one of these functions from step 7 to 11.

Class diagram:

Result:

Conclusion:

Experiment No. 3

Title: Write a program in C++ to implement Stack. Design the class for stack and the operations to be performed on stack. Use Constructors and destructors.

Aim: The objective of this assignment is to learn the concepts classes and objects, constructors and destructors.

Software required:

1. Turbo C++

Theory:

1. Constructors.
2. Destructors.

Algorithm:

1. Create a class as stack.
2. Declare the data members as x, top, stack [].
3. Declare the member function as
 - a) Stk ()
 - b) void push()
 - c) void pop()
 - d) void display()
4. Default constructor is used to initialize the value to zero.
5. In the push function.
 - a) Check the stack for overflow.
 - b) Increment the top value.
 - c) Read the value to be pushed to the stack.
 - d) Push the value into the stack.
 - e) Display the pushed value.
3. In the pop function.
 - a) Check the stack for underflow.
 - b) Pop the value from the stack.
 - c) Decrement the top value.
 - d) Display the value that is popped from the stack.
4. In the main, create the object for the class stk.

Class diagram:

Result:

Conclusion:

Experiment No. 4

Title: Write a program in C++ to perform following operations on complex numbers Add, Subtract, Multiply, Divide. Use operator overloading for these operations.

Aim: The objective of this assignment is to learn the concepts operator overloading.

Software required:

1. Turbo C++

Theory:

1. Operator overloading.
2. Function overloading.

Algorithm:

1. Define complex Class.
2. Define default constructor
3. Define conversion constructors
4. Declare other operator overloading functions
5. Define all the functions.
 - a) Addition:
 $(a+bi) + (x + yi) = ((a+x)+(b+y)i)$
 - b) Subtraction:
 $(a+bi) - (x + yi) = ((a-x)+(b-y)i)$
 - c) Multiplication:
 $(a+bi) * (x + yi) = (((a*x)-(b*y)) + ((a*y) + (x*b))i)$
 - d) Division:
i. $d=(x*x) + (y*y)$
ii. $(a+bi) / (x + yi) = (((a*x)+(b*y))/d) + (((b*x)-(a*y))/d)i$
6. Create objects for complex class in main() function.
7. The arithmetic operators will invoke the overloaded operator automatically and returns the result
8. Display the result using overloaded operators.

Class diagram:

Result:

Conclusion:

Experiment No. 5

Title: Write a program in C++ to implement database of persons having different profession e.g. engineer, doctor, student, laborer etc. using the concept of multiple inheritance.

Aim: The objective of this assignment is to learn the concepts of inheritance.

Software required:

1. Turbo C++

Theory:

1. Inheritance.
2. Multiple inheritances.

Algorithm:

1. Create the classes for engineer, doctor, student and laborer.
 - e) Class engineer
 - f) Class doctor
 - g) Class student
 - h) Class labor
2. Create derived class database.
3. In the main create an object to access the members of doctor and engineer class.

Class diagram:

Result:

Conclusion:

Experiment No. 6

Title: Write a program in Java to implement a Calculator with simple arithmetic operations such as add, subtract, multiply, divide, factorial etc. using switch case and other simple java statements.

Aim: The objective of this assignment is to learn Constants, Variables, and Data Types, Operators and Expressions, Decision making statements in Java.

Software required:

1. Java Development Kit.
2. Eclipse/Netbeans IDE.

Theory:

1. Constants, Variable and Data types of Java.
2. Operators and Expressions.

Algorithm:

1. Create a public class as a super class.
2. Create functions in the super class for:
 - a) Addition
 - b) Subtraction
 - c) Multiplication
 - d) Division
 - e) Factorial
3. In the main class create an object for the super class.
4. Using the switch statement perform the operation as per users choice.

Class diagram:

Result:

Conclusion:

Experiment No. 7

Title: Write a program in Java with class Rectangle with the data fields width, length, area and colour. The length, width and area are of double type and colour is of string type. The methods are get_length(), get_width(), get_colour() and find_area(). Create two objects of Rectangle and compare their area and colour. If the area and colour both are the same for the objects then display “ Matching Rectangles”, otherwise display “ Non-matching Rectangle”.

Aim: The objective of this assignment is to learn the concepts Classes, Objects and Methods of Java.

Software required:

1. Java Development Kit.
2. Eclipse/Netbeans IDE.

Theory:

1. Classes, objects and methods.

Algorithm:

1. Create the class rectangle.
2. Declare the data fields as private in the rectangle class.
 - a) private double width;
 - b) private double length;
 - c) private double area;
 - d) private String colour;
3. Function definitions for data fields.
4. In the main class create the two objects for area and color.
5. If the area and colour both are the same for the objects then display “ Matching Rectangles”, otherwise display “ Non-matching Rectangle”.

Class diagram:

Result:

Conclusion:

Experiment No. 8

Title: Write Programs in Java to sort i) List of integers ii) List of names.

Aim: The objective of this assignment is to learn Arrays and Strings in Java.

Software required:

1. Java Development Kit.
2. Eclipse/Netbeans IDE.

Theory:

1. Strings.
2. Vectors.

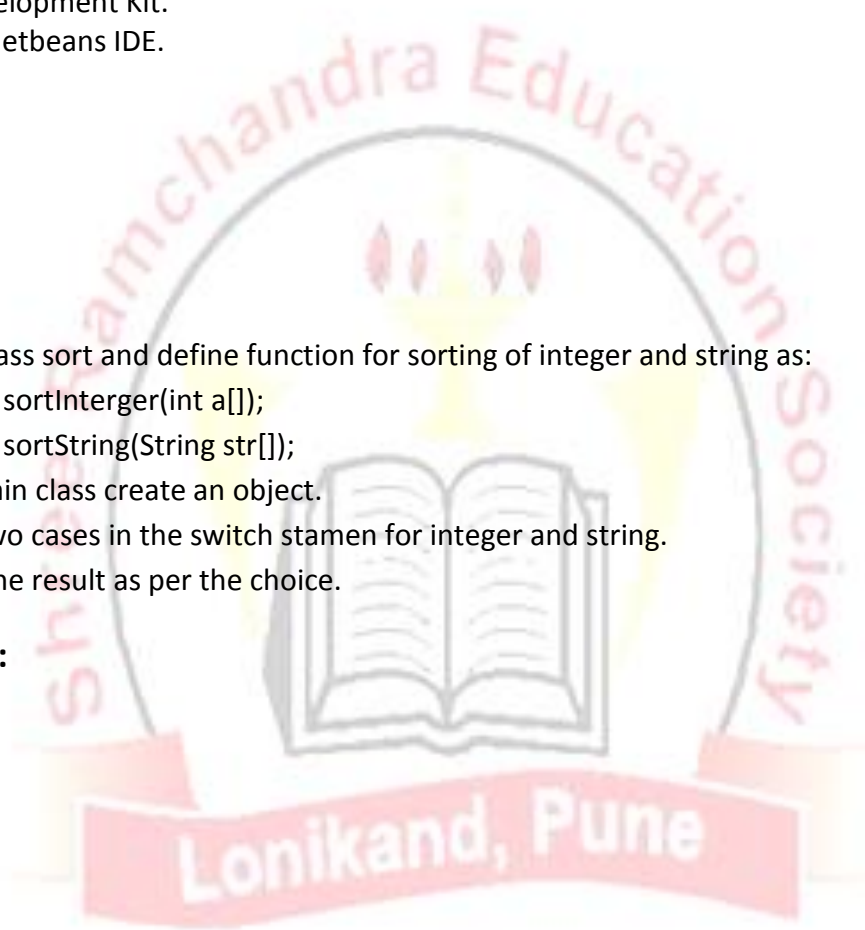
Algorithm:

1. Create class sort and define function for sorting of integer and string as:
e) void sortInteger(int a[]);
f) void sortString(String str[]);
2. In the main class create an object.
3. Create two cases in the switch stamen for integer and string.
4. Display the result as per the choice.

Class diagram:

Result:

Conclusion:



Experiment No. 9

Title: Write a Program in Java to add two matrices.

Aim: The objective of this assignment is to learn Arrays in Java.

Software required:

1. Java Development Kit.
2. Eclipse/Netbeans IDE.

Theory:

1. Arrays.

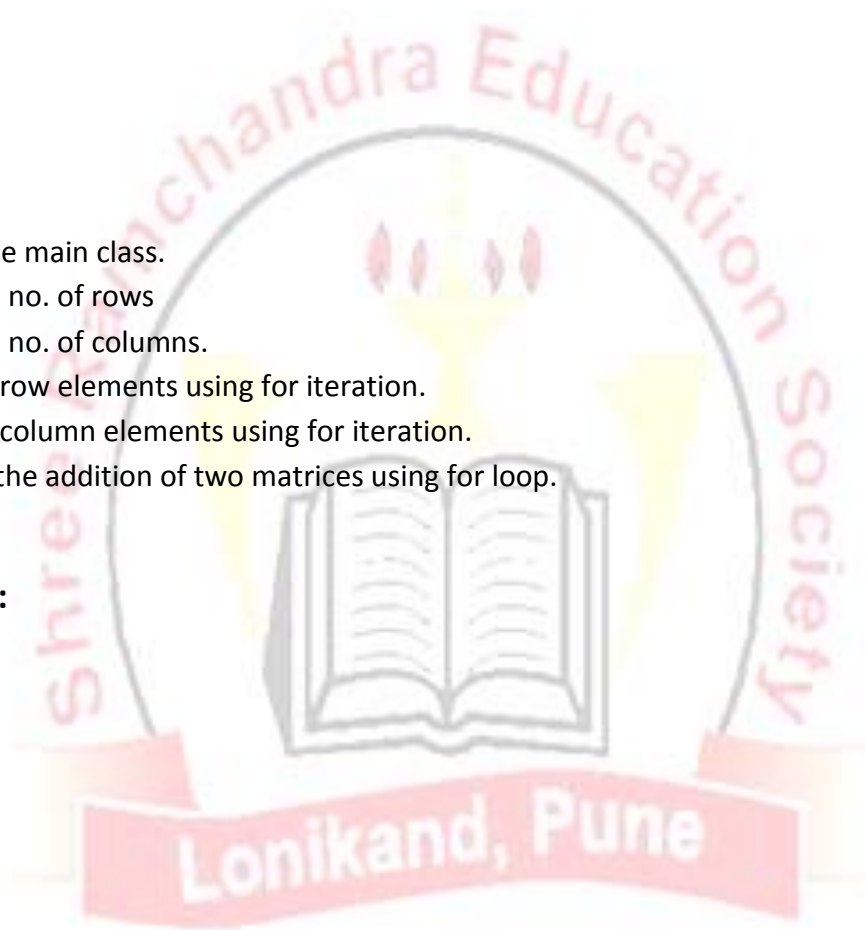
Algorithm:

1. Create the main class.
2. Enter the no. of rows
3. Enter the no. of columns.
4. Take the row elements using for iteration.
5. Take the column elements using for iteration.
6. Perform the addition of two matrices using for loop.

Class diagram:

Result:

Conclusion:



Experiment No. 10

Title: Write a program in Java to create a player class. Inherit the classes Cricket_player, Football_player and Hockey_player from player class.

Aim: The objective of this assignment is to learn the concepts of inheritance in Java.

Software required:

1. Java Development Kit.
2. Eclipse/Netbeans IDE.

Theory:

1. Interfaces.
2. Inheritance.

Algorithm:

1. Create class player.
2. Define the function display.
3. Inherits the classes from super class as:
 - g) class Cricket_Player extends Player
 - h) class FootBall_Player extends Player
 - i) class Hockey_Player extends Player

Class diagram:

Result:

Conclusion: