**Advanced Programming Laboratory**

**B.Tech. III Semester**



**Department: Computer Science and Engineering**

**Faculty of Engineering & Technology**

**M. S. Ramaiah University of Applied Sciences**

|  |  |
| --- | --- |
| Faculty | Engineering & Technology |
| Programme | B. Tech. in Computer Science and Engineering |
| Course | Advanced Programming Laboratory |
| Year/Semester | 2017/3rd Semester |
| Course Code | CSC207A |

List of Experiments

1. GUI design and Exception Handling
2. Simple application design and development
3. Introductory exercises in Java Classes, Interfaces and Constructors
4. Overloading, Inheritance and Overriding
5. Generic classes, Collections and their uses
6. Introductory exercises in Haskell
7. Types, Tuples, Lists and Strings, and Pattern Matching
8. Recursion and Sorting
9. Higher order functions and Folding

**General Instructions:**

Answer the following questions in a paragraph for respective sections:

**Documentation Section a:**

Explain how you moved from problem to solution. Write at least one core algorithm used. An algorithm should be simple English steps that can be implemented as a single function or method in the programming language. Each step should be a statement in the language.

**Documentation Section b:**

Conclusion must answer: Does the experiment achieve its aim and ILOs? Are there any limitations to constructs used in the experiment? What did you learn? Based on the experiment, what would you recommend to a software developer?

**Results and Discussions Section:**

Present at most 3 Screenshots with detailed explanation of each in the provided format.

# Index Sheet

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No.** |  |  |  |  | | **Lab Experiment** | **Viva**  **(6)** | **Results**  **(7)** | **Documentation**  **(7)** | **Total Marks**  **(20)** |
| 1 | GUI design and Exception Handling |  |  |  |  |
| 2 | Simple Application Design and Development |  |  |  |  |
| 3 | Introductory exercises in Java Classes, Interfaces and Constructors |  |  |  |  |
| 4 | Overloading, Inheritance and Overriding |  |  |  |  |
| 5 | Generic classes, Collections and their uses |  |  |  |  |
| 6 | Introductory Exercises in Haskell, Types, Tuples and Lists |  |  |  |  |
| 7 | Types, Tuples, Lists and Strings, and Pattern Matching |  |  |  |  |
| 8 | Recursion and Sorting |  |  |  |  |
| 9 | Higher Order Functions and Folding |  |  |  |  |
| 10 | Lab Internal Test conducted along the lines of SEE and valued for 50 Marks and reduced for 20 Marks | | | |  |
|  | **Total Marks** | | | |  |

**Component 1 (Lab Internal Marks) =**

**Signature of the Staff In-charge**

# Laboratory 1

Title of the Laboratory Exercise: GUI design and Exception Handling

1. Introduction and Purpose of Experiment

Students apply object oriented programming concepts for creating Graphical User Interfaces (GUIs).

1. Aim and Objectives

Aim

To apply object oriented programming concepts for creating Graphical User Interfaces

Objectives

At the end of this lab, the student will be able to

* Apply Object Oriented Programming to create Graphical User Interfaces
* Create stand-alone application user interfaces

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Java with C and Haskell programming languages. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Create a simple calculator using FXML in Java. Include exception handling logic that reports exceptions such as divide by zero in human readable pop-up messages.

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 2

Title of the Laboratory Exercise: Simple application design and development

1. Introduction and Purpose of Experiment

Students apply object oriented programming concepts for creating simple applications.

1. Aim and Objectives

Aim

To apply object oriented programming concepts for creating simple applications

Objectives

At the end of this lab, the student will be able to

* Apply Object Oriented Programming to create simple applications
* Create stand-alone applications in Java

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Java with C and Haskell programming languages. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Create a simple attendance management application using FXML in Java. Include exception handling logic that reports exceptions in human readable pop-up messages

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 3

Title of the Laboratory Exercise: Introductory exercises in Java Classes, Interfaces and Constructors

1. Introduction and Purpose of Experiment

Students apply object oriented programming to solve simple problems and get familiar with Java language

1. Aim and Objectives

Aim

To apply object oriented programming to solve simple

Objectives

At the end of this lab, the student will be able to

* Apply object oriented programming for solving simple problems
* Express solutions in Java language
* Use Greenfoot IDE

1. Experimental Procedure

For the problems listed below, design the algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Java with C and Haskell programming languages. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Write a program to develop a game for the scenario posed using Greenfoot IDE

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 4

Title of the Laboratory Exercise: Overloading, Inheritance and Overriding

1. Introduction and Purpose of Experiment

Students apply object oriented programming concepts including Overloading, Inheritance and Overriding to solve problems.

1. Aim and Objectives

Aim

To apply object oriented programming concepts including Overloading, Inheritance and Overriding to solve problems

Objectives

At the end of this lab, the student will be able to

* Apply Overloading, Inheritance and Overriding for solving problems
* Express solutions in Java language
* Use Netbeans IDE

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Java with C programming languages. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Write a program to develop a game for the scenario posed:

In ACME organization, there are many employees. All employees have first name, last name and aadhar number. ACME organization creates its own products and sells them. There are two types of sales employees: Commission employee are paid a percentage share (known as commission rate) of their gross sales. Base plus commission employee is a second type of sales employee who is paid a basic salary along with the commission. Other types of Employees include salaried employees who get paid a fixed weekly salary, piece workers who get paid a preset per piece amount based on the number of pieces they produce and hourly wage employees who get paid an hourly wage. Hourly wage employees also get 1.5 times the hourly wage for hours worked over 40 hours. Create a Java program to calculate salary of an employee in ACME organization.

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 5

Title of the Laboratory Exercise: Generic classes, Collections and their uses

1. Introduction and Purpose of Experiment

Students apply object oriented programming concepts Generic classes and Collections to solve problems.

1. Aim and Objectives

Aim

To apply object oriented programming concepts Generic classes and Collections to solve problems

Objectives

At the end of this lab, the student will be able to

* Apply Generic classes and Collections for solving simple problems
* Express generic solutions in Java language

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Java with C programming languages. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Write a program to solve for the following scenario:

Extend the program created in laboratory 2 to include logic for input of multiple employees. These employees should be stored in a collection such as ArrayList. The program should display the salaries of each employee as output, after all the employee details are entered. Extend the program further to display the employee details in ascending order of their salaries.

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 6

Title of the Laboratory Exercise: Introductory exercises in Haskell

1. Introduction and Purpose of Experiment

Students get familiar with the Haskell platform using a set of simple exercises.

1. Aim and Objectives

Aim

* To develop Haskell programs using GHCi

Objectives

At the end of this lab, the student will be able to

* Use GHCi to develop Haskell programs
* Develop Haskell programs edit and execute them successfully

1. Experimental Procedure

Students are given a set of instructions to be executed on the computer. The instructions should be edited and executed and documented by the student in the lab manual. They are expected to answer questions posed in section 5 based on their experiment.

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 7

Title of the Laboratory Exercise: Types, Tuples, Lists and Strings and Pattern Matching

1. Introduction and Purpose of Experiment

Students apply various basic data types and perform simple operations using patterns in Haskell

1. Aim and Objectives

Aim

* To apply correct data types and patterns for solving problems with functional programming

Objectives

At the end of this lab, the student will be able to

* Apply correct data type for given problem
* Apply pattern matching for function definition
* Solve problems using functional paradigm
* Manipulate lists

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Haskell with C programming language. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Find area of a triangle using Tuples and functions in Haskell

b. Filter only capital letters in a given string using functions and lists using Haskell

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 8

Title of the Laboratory Exercise: Recursion and Sorting

1. Introduction and Purpose of Experiment

Students apply recursion and mathematical functions to solve problems using functional programming paradigm

1. Aim and Objectives

Aim

To apply recursion and mathematical functions to solve problems using functional programming paradigm

Objectives

At the end of this lab, the student will be able to

* Apply recursion for solving problems
* Solve problems mathematically and express solution using functional paradigm

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Haskell with C programming language. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Create a program to do Quick Sort in Haskell

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |

# Laboratory 9

Title of the Laboratory Exercise: Higher order functions and Folding

1. Introduction and Purpose of Experiment

Students apply higher order programming to solve problems using functional programming paradigm

1. Aim and Objectives

Aim

To apply higher order programming to solve problems using functional programming paradigm

Objectives

At the end of this lab, the student will be able to

* Apply higher order programming for solving problems
* Express functions in terms of other functions

1. Experimental Procedure

For the problems listed below, design the data structures, algorithm(s) and write the program(s). Tabulate the output for various inputs and verify against expected values. Compare the programming method in Haskell with C programming language. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome.

a. Find if a given list is palindrome using strings and higher order functions in Haskell

|  |  |
| --- | --- |
| **Documentation:** | |
| a. Procedure and Algorithm(s): |  |
| b. Conclusions : |  |

|  |
| --- |
| **Results and Discussions:** |
| **Screenshot:** |
| **Discussion:** |