**Computer Simulation Laboratory**

**B.Tech. 5th Semester**

****

**Department: Computer Science and Engineering**

**Faculty of Engineering & Technology**

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



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

Private University Established in Karnataka State by Act No. 15 of 2013

|  |  |
| --- | --- |
| Faculty | Engineering & Technology |
| Programme | B. Tech. in Computer Science and Engineering |
| Year/Semester | 5th Semester |
| Name of the Laboratory | Computer Simulation Laboratory |
| Laboratory Code | CSC308A |

List of Experiments

1. Introduction to Java simulation
2. Implementing a Java program for random numbers generation
3. Implementing a Java program for testing random numbers generation
4. Simulation of a single server queue
5. Experiments on single server queue
6. Experiments on single server queue
7. Simulation of Multiple server queue
8. Experiments on Multiple server queue
9. Experiments on Multiple server queue

# Laboratory 1

Title of the Laboratory Exercise: Introduction to java simulation

1. Introduction and Purpose of Experiment

Computer simulation provides students to design and implement computer simulation models, conduct simulation experiments and evaluate system performance. This laboratory exercise will help the students to get familiar with using object‐oriented simulation in Java.

Java (Structured Parallel Discrete Event Simulation in Java) system is designed to incorporate the parallel programming technology into discrete event simulations. The java system adopts the approach of augmenting a general-purpose language with essential constructs to support simulation modeling based on the process-oriented modeling technology

1. Aim and Objectives

Aim

* To use Netbeans and understand using object‐oriented simulation in Java

Objectives

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

* Explain the features and use of Netbeans IDE to develop java programs for simulation
* Edit, compile and execute java programs successfully using Netbeans IDE

1. Experimental Procedure

Students are given a set of programs for generating random numbers using built-in methods. Programs should be edited, compiled and executed using Netbeans IDE.

Random number generation using inbuilt methods/manually

Ex: coin toss, die, and cards

1. Calculations/Computations/Algorithms
   1. Develop and implement a java program to generate 10 random number
   2. Develop and implement a java program to generate 10 random numbers in between 0 to 1
   3. Develop and implement a java program to generate 10 normally distributed random numbers in between 0 to 1.
   4. Develop and implement a java program to flip a coin 50 time, and count number of occurrence of head and tail and determine probability distribution of head and tail.
   5. Develop and implement a java program to through a dice 200 times, and count number of occurrence of each face (1,2 …. 6) and determine probability distribution.
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

# Laboratory 2

Title of the Laboratory Exercise: Random Number Generation

1. Introduction and Purpose of Experiment

Random numbers are widely used ingredient in the simulation of almost all discrete systems. Simulation languages generate random numbers that are used to generate event times and other random variables. Random number generators have applications in gambling, statistical sampling, computer simulation, cryptography, completely randomized design and other areas where producing an unpredictable result is desirable. The generation of pseudo random numbers is an important and common task in computer programming.

1. Aim and Objectives

Aim

* To develop programs generating random numbers

Objectives

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

* Use different random generation methods for generating random numbers
* Create java programs for generating random numbers.

1. Experimental Procedure
   * 1. Analyse the problem statement
     2. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
     3. Implement the algorithm in java language
     4. Compile the java program
     5. Test the implemented program
     6. Document the Results
     7. Analyse and discuss the outcomes of your experiment
2. Questions

Implement the following in java

* 1. Generate pseudorandom numbers are based on the linear congruential random number generator invented by Lehmer
  2. [Generate random numbers from the standard uniform distribution.](http://in.mathworks.com/help/stats/generate-random-numbers-using-the-uniform-distribution-inversion-method.html#zmw57dd0e14873)
  3. [Generate random numbers from the standard normal distribution.](http://in.mathworks.com/help/stats/generate-random-numbers-using-the-uniform-distribution-inversion-method.html#zmw57dd0e14930)

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

# Laboratory 3

Title of the Laboratory Exercise: Testing Random Number Generators.

1. Introduction and Purpose of Experiment

Random numbers are needed that are distributed according to a prescribed distribution. In the first example the distribution is uniform. Each random number has the same a priori probability. Randomness tests in data evaluation, are used to analyse the distribution of a set of data to see if it is [random](https://en.wikipedia.org/wiki/Random). In [stochastic modelling](https://en.wikipedia.org/wiki/Stochastic_modeling), as in [computer simulations](https://en.wikipedia.org/wiki/Computer_simulation), randomness of potential input data can be verified, using different statistical methods, to show that the data are valid for use in simulation runs.

2. Aim and Objectives

Aim

* To develop programs to test the Randomness of a Random Numbers Generated

Objectives

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

* Understand different statistical tests devised to check for the properties of random number generators
* Understand how well the empirical distribution, i.e., the generated sequence, fits a test distribution
* Develop java program to test randomness of generated numbers.

1. Experimental Procedure
   * 1. Analyse the problem statement
     2. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
     3. Implement the algorithm in java language
     4. Compile the java program
     5. Test the implemented program
     6. Document the Results
     7. Analyse and discuss the outcomes of your experiment
2. Questions

Create java program to implement

1. Chi-squared test for random number generation
2. Kolmogorov-Smirnov test for random number generation
3. Calculations/Computations/Algorithms
4. Presentation of Results
5. Analysis and Discussions
6. Conclusions
7. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

# Laboratory 4, 5, 6

Title of the Laboratory Exercise: Simulation of Single Server Queuing System

1. Introduction and Purpose of Experiment

This lab experiment shows how to model a single-queue single-server system with a single traffic source and an infinite storage capacity. In the notation, the M stands for Markovian; M/M/1 means that the system has a Poisson arrival process, an exponential service time distribution, and one server. Queuing theory provides exact theoretical results for some performance measures of an M/M/1 queuing system and this model makes it easy to compare empirical results with the corresponding theoretical results.

**Structure**

The model includes the components listed below:

* Time Based Entity Generator block: It models a Poisson arrival process by generating entities (also known as "customers" in queuing theory).
* Exponential Interarrival Time Distribution subsystem: It creates a signal representing the interarrival times for the generated entities. The interarrival time of a Poisson arrival process is an exponential random variable.
* FIFO Queue block: It stores entities that have yet to be served.
* Single Server block: It models a server whose service time has an exponential distribution.

2. Aim and Objectives

Aim

* To develop programs to Simulation of Single Server Queuing System

Objectives

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

* Simulate different problems on Single Server Queuing System
* Create Java simulation class hierarchy by implementing sub-classes of the Thread class to simulate simple M/M/1 queue

1. Experimental Procedure
   * 1. Analyse the problem statement
     2. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
     3. Implement the algorithm in java language
     4. Compile the java program
     5. Test the implemented program
     6. Document the Results
     7. Analyse and discuss the outcomes of your experiment
2. Questions

Implement a Simulation of Single Server Queuing System for given problem

Structure of model should include:

1. Time Based Entity Generator
2. Exponential Interarrival Time Distribution subsystem.
3. FIFO Queue
4. Single Server
5. Calculations/Computations/Algorithms
6. Presentation of Results
7. Analysis and Discussions
8. Conclusions
9. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

# Laboratory 7, 8, 9

Title of the Laboratory Exercise: Simulation of Two-Server Queuing System

1. Introduction and Purpose of Experiment

This lab shows how to model a single-queue multiple-server system. In the notation, the M stands for Markovian; M/M/s means that the system has a Poisson arrival process, an exponential service time distribution, and **s** (multiple) server. Queuing theory provides exact theoretical results for some performance measures of an M/M/s queuing system and this model makes it easy to compare empirical results with the corresponding theoretical results.

In simulating real world systems on computer like a Multi-channel queue or Able Baker problem consisting of a Two servers for serving the two queues of customers.so we have its following simulation along with the able baker problem with it.

**Structure**

The model includes the components listed below:

* Time Based Entity Generator block: It models a Poisson arrival process by generating entities (also known as "customers" in queuing theory).
* Exponential Interarrival Time Distribution subsystem: It creates a signal representing the interarrival times for the generated entities. The interarrival time of a Poisson arrival process is an exponential random variable.
* FIFO Queue block: It stores entities that have yet to be served.
* Multiple Server block: It models a servers whose service time has an exponential distribution.

1. Aim and Objectives

Aim

* To develop concurrent programs for Simulation of Two-Server Queuing System

Objectives

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

* Simulate different problems on multiple Server Queuing System
* Create Java simulation class hierarchy by implementing sub-classes of the Thread class to simulate simple M/M/s queue

1. Experimental Procedure
   * 1. Analyse the problem statement
     2. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
     3. Implement the algorithm in java language
     4. Compile the java program
     5. Test the implemented program
     6. Document the Results
     7. Analyse and discuss the outcomes of your experiment
2. Questions

Implement a Simulation of Multiple Server Queuing System for given problem

Structure of model should include:

* 1. Time Based Entity Generator
  2. Exponential Interarrival Time Distribution subsystem.
  3. FIFO Queue
  4. Multiple Server

1. Calculations/Computations/Algorithms
2. Presentation of Results
3. Analysis and Discussions
4. Conclusions
5. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations