1. a) How simulation plays an important role in new innovation?[3]

Simulation plays a **crucial role in driving new innovation** because it allows researchers, engineers, and designers to **experiment, predict, and improve ideas** in a virtual environment before implementing them in the real world. Here's a clear explanation of its importance:

1. Reduces Cost and Risk

- Simulation lets innovators **test prototypes virtually**, avoiding the expense of building multiple physical models.
- It helps identify design flaws and safety risks early, saving both time and money.

Example: In aerospace, flight simulators test new aircraft designs before building real ones.

2. Accelerates Innovation

- Innovators can quickly **try different design variations** and analyze outcomes without waiting for real-world tests.
- This speeds up **research and development cycles** and enables faster product improvement.

3. Enables Complex Problem Solving

- Many systems (like climate models, autonomous vehicles, or microchips) are too complex to test physically.
- Simulation helps understand how these systems behave under various conditions, leading to **smarter and more reliable designs**.

4. Encourages Creativity and Exploration

- Since failure in simulation carries no real-world consequences, it encourages **bold experimentation**.
- Innovators can explore unconventional ideas that might be too risky or expensive to test physically.

5. Enhances Decision-Making

- By providing data-driven insights and predictive analytics, simulation helps decision-makers **evaluate different strategies** and choose the best option.

6. Improves Training and Real-World Readiness

- Simulation environments train people to handle new technologies safely and effectively before actual deployment.
  Example: Surgeons use VR simulations to practice complex operations before performing them on patients.

b)In which situations simulation is not a better choice?[3]

Simulation is **not always the best choice** in every situation. It has limitations and may be unsuitable in the following cases:

1. When Real Data or Models Are Unavailable

- If there is **insufficient real-world data** or understanding of the system, the simulation results may be inaccurate or misleading.
  Example: Simulating a new disease outbreak without enough biological data can lead to wrong predictions.

2. When High Accuracy Is Required

- Simulations are **approximations**, not exact representations.
- In critical systems (like nuclear reactors or medical implants), even a small error can have severe consequences — so **real testing** is essential.

3. When Cost of Simulation Is Too High

- Developing complex simulations (e.g., for aircraft aerodynamics or weather systems) requires **expensive software, hardware, and expertise**.
- For small projects, it may not be cost-effective compared to real-world experiments.

4. When Human Behavior Plays a Major Role

- Simulating **human emotions, intuition, and decision-making** is difficult.
  Example: Simulating customer reactions to a new product may not reflect real human responses accurately.

5. When Validation Is Difficult

- If the simulation cannot be compared with **real-world results**, its accuracy cannot be verified — reducing trust in the outcome.

6. When Time Constraints Exist

- Building and testing a detailed simulation can take **a long time**.
- For urgent decisions, simpler analytical or empirical methods might be more practical.

c) What are different methods of studying a system before designing a simulation?[3]

Different methods of studying a system before designing a simulation include:

1. **Observation:** Directly watching how the system operates in real life to understand its processes, interactions, and behavior.
2. **Interviews and Questionnaires:** Collecting information from people who are involved with or knowledgeable about the system, such as operators, managers, or users.
3. **Document Review:** Studying existing records, reports, manuals, and performance data to understand how the system currently works.
4. **Data Collection and Analysis:** Gathering quantitative data about inputs, outputs, and performance to identify patterns or problems.
5. **Process Mapping or Flowcharting:** Creating diagrams to visualize the steps, decisions, and flow of information or materials within the system.

6. **Experimentation:** Making small controlled changes in the real system to observe how it reacts, which helps in understanding system dynamics.
7. **Mathematical or Analytical Modeling:** Using equations and formulas to represent the relationships among system variables before developing a simulation model.

.

d)Classify simulation models

Simulation models can be classified in several ways based on **time, randomness, and representation**. The main classifications are:

1. **Based on Time Handling:**
   - **Continuous Simulation:** The system changes continuously over time.
     Example: Water flow in a pipeline or chemical reactions.
   - **Discrete Event Simulation:** The system changes only at specific points in time when events occur.
     Example: Customers arriving at a bank or packets in a network.
2. **Based on Randomness:**
   - **Deterministic Simulation:** No randomness is involved; the same inputs always produce the same outputs.
     Example: Projectile motion under fixed conditions.
   - **Stochastic Simulation:** Includes random variables; results may vary even with the same inputs.
     Example: Traffic flow or weather forecasting.
3. **Based on Model Representation:**
   - **Physical Simulation:** Uses physical objects to represent the real system.
     Example: Wind tunnel testing for aircraft models.

- **Computer Simulation (Mathematical Simulation):** Uses mathematical models and computer programs to mimic real systems.

4.Static vs. Dynamic:
- Static: Represents a system at a single point in time. Time does not play a role.
- Dynamic: Represents how a system evolves over time.

2.a.Briefly explain the differences between discrete system and continuous system with suitable figures [4] repeat 7th batch final

b.Discuss the concept of "Time Advance Mechanism" with an example.[4] **repeat 7th batch 1st mid**

c.What are different components of a Discrete-event simulation model? Explain.[4] repeat 7th batch final

3. a. Define Chi-Square Test.[2]

Chi-Square (χ^2) Test

The **Chi-Square Test** is a statistical hypothesis test used to determine whether there is a **significant difference between observed frequencies and expected frequencies**, or to check whether **two categorical variables are related**.

It compares actual data with theoretical or expected data to see if any observed difference occurs by chance.

$$\chi_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Where:

- O_i = Observed frequency in the i-th class interval.
- E_i = Expected frequency in the i-th class interval.
- k = Number of class intervals.

If the calculated χ_0^2 is less than the critical value $\chi^2_{\alpha,k-s-1}$ from the table, we accept the hypothesis that the data follows the specified distribution.

b) Generate three Poisson variants with mean alpha=0.2 using acceptance rejection techniques. Given random numbers are 0.4357, 0.4146, 0.8353, 0.9952 and 0.8004n [4] repeat 7th batch pdf

c.The life of a device used to inspect cracks in aircraft wings is given by X, a continuous random variable assuming all values in the range x>=0. The cdf of the device's lifetime, in years, is as follows-[6] repeat 7th batch pdf

$$F(x) = \frac{1}{2}\int_0^x e^{-t/2}dt$$

i.Find the probability that the device will last for <2 years
ii.Find the probability that the device will last between 2 and 3 years.

4.a.Explain over all aspects and characteristics of a Queueing System[8]

A queuing system is defined by the interaction between incoming flow (arrivals), the waiting line (queue), and the processing facility (servers). The system is characterized by the following eight core aspects:

1. Arrival Process (The Input Source): This describes how customers or jobs enter the system.

- **Pattern:** Arrivals can be **Deterministic** (fixed intervals) or **Stochastic** (random intervals).
- **Distribution:** Random arrivals are often modeled using a **Poisson process**, where inter-arrival times follow an **Exponential distribution**.
- **Parameter:** The mean arrival rate is denoted by \lambda (lambda).

2. Service Mechanism: This defines the nature of the service provided.

- **Duration:** Service time can be constant (automated machines) or random (human tellers).
- **Parameter:** The mean service rate is denoted by \mu (mu).
- **Arrangement:** Servers can be arranged in **Parallel** (multiple tellers, one line) or in **Series** (assembly line stages).

3. Queue Discipline: The logic or rule used to select the next entity for service.

- **FIFO (First-In, First-Out):** The standard "fair" queue (e.g., a checkout line).
- **LIFO (Last-In, First-Out):** Common in inventory stacks (e.g., plates).
- **Priority:** High-urgency jobs are processed first regardless of arrival time (e.g., Emergency Room).

4. System Capacity: The physical limitation on the number of entities allowed in the system (queue + service) at one time.

- **Infinite Capacity:** No limit on the line length; every arrival enters.
- **Finite Capacity:** There is a limit (e.g., a buffer of size K). If the system is full, new arrivals are blocked or lost.

5. Number of Servers: The capacity of the service facility to handle simultaneous customers.

- **Single-Server:** One channel handles all traffic (e.g., M/M/1).
- **Multi-Server:** Multiple channels work simultaneously (e.g., M/M/c).
- *Note:* Increasing servers reduces waiting time (W_q) but increases operational costs.

6. Queue Behavior: In systems involving humans, psychological reactions can alter the queue dynamics:

- **Balking:** A customer sees a long line and refuses to join it.
- **Reneging:** A customer joins the line but leaves before being served due to impatience.
- **Jockeying:** A customer moves from a slow line to a faster line in a multi-queue system.

7. Population Source: The size of the potential pool of customers.

- **Infinite Population:** The arrival of one customer does not affect the probability of future arrivals (Standard assumption).
- **Finite Population:** The pool is limited (e.g., 5 machines in a factory). If a machine is broken, the "arrival rate" of future breakdowns decreases.

8. Performance Measures: The quantitative metrics used to evaluate the efficiency of the system:

- **L_q:** Average number of customers in the queue.

- **L:** Average number of customers in the system.
- **W_q:** Average waiting time in the queue.
- **W:** Average time spent in the system.
- **ρ (Rho):** Server utilization factor ($\rho = \lambda/\mu$).

b. Customers at a restaurant arrive in groups (one to eight persons). The number of persons (per group) for 300 customers and the relative frequencies appear are shown below, Draw empirical CDF.

Arrivals per party	1	2	3	4	5	6	7	8
Frequency	30	110	45	71	12	13	7	12

5.a,Explain different methods of random number generation.[9]

Here is a compact, high-scoring answer covering the core methods and additional techniques for [9 Marks].

5(a). Methods of Random Number Generation

Random number generation in simulation relies on **Pseudo-Random Number (PRN)** algorithms. These algorithms produce a sequence of numbers that are deterministic (calculated via formulas) but satisfy statistical tests for **Uniformity** and **Independence**.

Here are the six primary methods:

1. Mid-Square Method

The earliest method, proposed by Von Neumann.

- **Process:** Take a 4-digit seed (X_0), square it (X_0^2), and extract the **middle 4 digits** to form the next number (X_1).
- **Formula:** $X_{i+1} = \text{middle digits of } (X_i^2)$
- **Drawback:** It has a tendency to degenerate (reach 0000) very quickly.

2. Linear Congruential Method (LCM)

The most widely used method due to its speed and simplicity. It produces a sequence of integers X between 0 and $m - 1$.

- **Formula:**

$$X_{i+1} = (aX_i + c) \mod m$$

  - X_0: Seed (Initial value)
  - a: Constant Multiplier
  - c: Increment
  - m: Modulus

- **Normalization:** To get a random number R_i between $[0, 1]$, divide by m:

$$R_i = \frac{X_i}{m}$$

3. Multiplicative Congruential Method

A variation of the LCM where the increment c is set to zero. It is computationally faster but requires a prime modulus m to achieve a good cycle length.

- **Formula:**

$$X_{i+1} = (aX_i) \mod m$$

4. Additive Congruential Method

This method generates the next number by adding two previously generated numbers (often with a lag k) rather than multiplying the current one.

- **Formula:**

$$X_{i+1} = (X_i + X_{i-k}) \mod m$$

- **Advantage:** It can produce longer periods (cycles) than simple LCMs.

5. Combined Linear Congruential Generators (CLCG)

Since single LCGs have a limited period (cycle length), this method runs multiple LCGs simultaneously and combines their results mathematically.

- **Concept:** If $X_{i,1}$ is from Generator 1 and $X_{i,2}$ is from Generator 2:

$$X_{new} = (X_{i,1} - X_{i,2}) \mod (m_1 - 1)$$

- **Advantage:** Creates a massively long period (e.g., 10^{18}), making it ideal for modern, complex simulations.

6. Tausworthe (Feedback Shift Register) Method

Instead of integer arithmetic, this method operates on **binary bits**.

- **Process:** It generates a sequence of bits (0s and 1s) using a recurrence relation involving the Exclusive-OR (XOR) operation. These bits are then grouped to form integers.
- **Advantage:** Very fast on digital computers because it uses hardware-level bitwise operations.

b.What are the challenges in generating pseudo random numbers?[3]

Challenges in Generating Pseudo-Random Numbers

Since pseudo-random numbers are generated by mathematical algorithms (finite state machines), they face specific structural limitations:

1. **The Period (Cycle Length) Limit**
   Because computers have a finite number of states, every random number generator will eventually repeat

its sequence.1 The challenge is ensuring the Period is large enough so that the repetition does not occur within the lifespan of the simulation run.

2. **Statistical Validity (Uniformity & Independence)**
   It is difficult to create a purely deterministic sequence that lacks pattern or correlation. Generators must be rigorously tuned to ensure numbers don't "cluster" in certain ranges (bad uniformity) or depend on previous values (autocorrelation), which would bias simulation results.2
3. **Parameter Selection**
   In methods like the Linear Congruential Generator (aX+c mod m), choosing the wrong constants (a, c, m) can lead to disastrously short cycles or obvious patterns.5 Finding "safe" parameters that work for all seed values is mathematically complex.

6.a.When Linear Congruential Generator is used?[2]

The Linear Congruential Generator is used when:

1. **Computational Efficiency is Required:** It is extremely fast and uses very little memory, making it ideal for simulations that need to generate millions of random numbers quickly without slowing down the system.[1]
2. **Simplicity is Sufficient:** It is used in general-purpose simulations (like basic queuing systems or educational models) where cryptographic security is not required, and simple statistical uniformity is enough.

b.How is the Linear Congruential Generator implemented? Explain with an example.[4]

Implementation of Linear Congruential Generator (LCG)

The Linear Congruential Generator (LCG) is implemented using a recursive mathematical formula that generates a sequence of pseudo-random integers. Each new number is determined entirely by the previous number.

1. The Formula: The sequence of integers X_1, X_2, ……is generated using:

$$X_{i+1} = (aX_i + c) \mod m$$

To convert these integers into random numbers R_i between 0 and 1 (which are used in probabilities), we divide by the modulus:

$$R_i = \frac{X_i}{m}$$

2. Key Parameters: To implement this, you must define four constant values:

- X_0: The **Seed** (starting value).
- a: The **Multiplier**.
- c: The **Increment**.
- m: The **Modulus** (determines the maximum value).

Example: Let's implement a simple LCG with the following parameters:

- **Seed (X_0):** 27
- **Multiplier (a):** 17
- **Increment (c):** 43
- **Modulus (m):** 100

Step 1: Generate the first number (X_1)

X_1 = (17 *27 + 43) mod 100

X_1 = (459 + 43) mod 100

X_1 = 502 mod 100 = 2

Random Number (R_1): 2 / 100 =0.02

Step 2: Generate the second number (X_2)

Use the result from Step 1 ($X_1 = 2$) as the input.

$X_2 = (17 * 2 + 43) \mod 100$

$X_2 = (34 + 43) \mod 100$

$X_2 = 77 \mod 100 = 77$

Random Number (R_2): 77 / 100 = 0.77

Conclusion: The sequence starts with integers **2, 77**, corresponding to random values **0.02, 0.77**.

c.Explain two methods of uniformity test with appropriate examples.[6].

1. The Chi-Square (X^2) Test

This test is best used for **large sample sizes** (N >= 50). It checks if the random numbers are distributed equally across the range [0, 1].

The Procedure:

1. Divide the range [0, 1] into k equal sub-intervals (bins).
2. Count how many generated numbers fall into each interval (Observed Frequency, O_i).
3. Calculate the Expected Frequency (E_i). For a uniform distribution, $E_i = \frac{N}{k}$.
4. Calculate the test statistic:

$$\chi_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

5. Compare X_0^2 with the critical value from the table. If $X_0^2 < X_{critical}^2$, the numbers are Uniform.

Example:

1. **Data:** We generate N=100 random numbers.
2. **Intervals:** We divide [0, 1] into k=10 intervals.

3. **Expected Frequency:** We expect 100/10 = 10 numbers in each interval.
4. **Observation:** In the first interval [0.0, 0.1), we observe O_1 = 12 numbers.
5. Calculation for Interval 1:

$$\frac{(12-10)^2}{10} = \frac{4}{10} = 0.4$$

6. Sum this calculation for all 10 intervals to get the final X_0^2.

---

2. The Kolmogorov-Smirnov (K-S) Test

This test is best used for **small sample sizes** (N < 50). It compares the continuous Cumulative Distribution Function (CDF) of the uniform distribution with the empirical CDF of the sample data.

The Procedure:

1. **Rank** the data points from smallest to largest: $R_{(1)} \leq R_{(2)} \cdots \leq R_{(N)}$.
2. Calculate the deviation between the position of the number and its value using two formulas:
    - $D^+ = \max\left(\frac{i}{N} - R_{(i)}\right)$
    - $D^- = \max\left(R_{(i)} - \frac{i-1}{N}\right)$
3. Find the maximum deviation D = max(D^+, D^-).
4. If D is less than the critical value D_α from the K-S table, the hypothesis is accepted.

Example:

- **Data:** 5 numbers: 0.05, 0.14, 0.44, 0.81, 0.93. (N=5).
- **Step for the 2nd number (i=2, Value = 0.14):**
    - Theoretical position should be 2/5 = 0.40.
    - Difference: 0.40 - 0.14 = 0.26.

- We repeat this for all 5 numbers. The largest difference found is our test statistic D.

7.a.Explain how simulation and modelling can play an important role in Manufacturing and Material Handling Systems.[3]

Simulation is vital for optimizing production efficiency and minimizing costs. Key roles include:

1. **Identifying Bottlenecks:** Pinpoints constraints where work piles up, allowing for targeted process improvements.
2. **Resource Optimization:** Determines the exact number of machines, workers, and forklifts needed to meet demand without waste.
3. **Layout Validation:** Tests different factory floor arrangements (e.g., cell vs. line layout) to minimize material travel time.
4. **Production Scheduling:** Evaluates different job sequences and shift patterns to find the most efficient schedule.
5. **Inventory Management:** helps balance Work-In-Process (WIP) levels to prevent stockouts or excessive storage costs.
6. **Failure Analysis:** Predicts the impact of machine breakdowns on overall system throughput.

b.Briefly describe probable simulation processes in a Manufacturing System [Use an appropriate example).[6]

In a manufacturing simulation, the goal is to model the flow of a product from raw material to finished good. The simulation logic typically involves tracking entities (parts/jobs) as they move through specific state-changing processes.

Here are the key processes typically modeled:

1. **Arrival Process (Source):** The generation of raw materials or sub-components entering the system. This is modeled using an inter-arrival time distribution (e.g., a pallet of metal sheets arrives every 10 minutes).

2. **Buffering/Queuing:** If a machine is busy, parts must wait in a buffer (storage area). The simulation tracks queue length and waiting time to identify bottlenecks.
3. **Processing (Service):** The actual work done on the part (e.g., drilling, painting, assembly). This consumes time and resources. Service times are usually stochastic (random) due to operator variance.
4. **Material Handling (Transport):** The movement of parts between workstations using conveyors, forklifts, or AGVs (Automated Guided Vehicles).3 This adds transit time to the total cycle time.4
5. **Breakdowns and Maintenance:** Machines are not always available. The simulation must model Downtime events based on "Time Between Failures" and "Time to Repair" to see how breakdowns stop the production flow.

Example: A Drilling Center Simulation

Imagine a simple manufacturing cell with one **Drill Press** and a **Conveyor Belt**.

- **Step 1 (Arrival):** Raw metal plates arrive at the station every 5 (+-) 1 minutes (Uniform distribution).
- **Step 2 (Queue):** The plates enter an input buffer. If the drill is busy, the Queue Count (L_q) increases.
- **Step 3 (Processing):** The drill operator picks a plate. The drilling operation takes roughly 4 minutes (Normal distribution).
- **Step 4 (Breakdown - Random Event):** Every 100 hours, the drill bit breaks. The simulation pauses production for 30 minutes while a technician fixes it.
- **Step 5 (Transport):** Once drilled, the plate is placed on a conveyor belt which takes 2 minutes to move it to the inspection station.

Analysis: By running this simulation, the factory manager can determine if one drill is enough or if a second drill is needed to prevent the input buffer from overflowing.

c.Define verification in simulation process? Describe techniques to perform verification on simulation models.[3]

Definition: Verification is the process of determining if the model implementation accurately represents the developer's conceptual description and specifications.

- It asks the question: **"Are we building the product right?"**
- It focuses on **debugging**—ensuring the computer code (program) runs error-free and the logic (loops, conditions) matches the flowchart.

Techniques to Perform Verification:

1. **Structured Walk-through:** A peer or supervisor reviews the code line-by-line to check for logic errors or incorrect assumptions without running the program.
2. **Trace Analysis:** The simulation is run, and the state of variables (like Queue Length or Clock Time) is printed out after every single event. This log is compared manually against hand-calculated results to ensure the math is correct.
3. **Simplified Cases:** Run the model under extreme or simple conditions where the answer is known.
   - *Example:* If the arrival rate is 0, the total customers served should be 0. If this doesn't happen, the code is wrong.
4. **Animation:** Using the simulation software's visualizer to watch the entities move. If you see a box moving through a wall or a customer skipping a line, it indicates a logic error in the code.

8 . a)Monte Carlo simulation is a special case of stochastic simulation? Comment.[3]

Comment: Yes, Monte Carlo simulation is considered a special, static case of stochastic simulation.

Here is the reasoning:

1. **Stochastic Nature (The Similarity):** Like all stochastic simulations, Monte Carlo methods rely heavily on random numbers and probability distributions to generate output.1 It does not produce the exact same result every time it is run.2
2. **The "Special" Distinction (Static vs. Dynamic):**
   - **General Stochastic Simulation (e.g., DES):** Is typically **dynamic**. It tracks how a system changes *over time* (e.g., a queue growing and shrinking hour by hour).
   - **Monte Carlo Simulation:** Is typically **static**. It takes a "snapshot" of a problem. It uses random sampling to solve mathematical problems (like calculating the area under a curve or estimating risk) where time is not a variable.[3]

Conclusion: Monte Carlo is "stochastic" because it uses randomness, but it is a "special case" because it typically solves problems that do not evolve over time.

b) What method did we use to develop the model for the ATM (bank) case study?[3]

To develop the model for the ATM case study, we used the **Event-Scheduling Approach** of Discrete-Event Simulation (DES).

This method involves three key steps:

1. **Defining Events:** We identified the specific instants that change the system's state:
   - **Arrival Event:** A customer arrives at the ATM.
   - **Departure Event:** A customer finishes their transaction and leaves.
2. **Time Advance:** We used the **Next-Event Time Advance** mechanism. The simulation clock does not tick second-by-second; instead, it "jumps" directly to the time of the next scheduled event.
3. **State Variables:** We tracked the system state (e.g., LQ = number in queue, LS = status of ATM) only at these discrete event times.

c) What is SIMSCRIPT? What is its importance? Discuss its various functions[3]

What is SIMSCRIPT?
SIMSCRIPT is a powerful, high-level simulation programming language designed specifically for building Discrete-Event Simulation (DES) models.1 It is English-like in syntax, making it readable and easy to structure.2

Importance:
It is important because it bridges the gap between general-purpose languages (like C++ or Java) and rigid simulation software. It provides built-in tools for handling time and lists, saving developers from writing complex "clock-advancing" code from scratch, while still offering the flexibility to model very complex logic.

Various Functions:

1. **Time Flow Mechanism:** It automatically manages the simulation clock and the "Future Event List," ensuring events happen in the correct order.
2. **Entity & Set Management:** It has built-in commands to create entities (e.g., CREATE A CUSTOMER) and manage sets/queues (e.g., FILE CUSTOMER IN QUEUE), making it easy to model waiting lines.
3. **Statistical Collection:** It automatically accumulates data (like means, variances, and histograms) for specified variables without requiring manual calculation code.

d) Write a short note on Statistical output analysis.[3]

Definition: Statistical output analysis is the process of examining the data generated by a simulation (e.g., waiting times, queue lengths) to estimate the true performance of the system. It ensures that the results are not just random noise from a single run but are statistically valid.

Key Challenges:
Unlike simple statistical experiments, simulation output data is often:

1. **Non-Stationary:** The system behavior changes over time (e.g., a queue filling up at the start).
2. **Autocorrelated:** Data points are not independent (e.g., if Customer 1 waits a long time, Customer 2 is also likely to wait a long time).

Types of Analysis:

- **Terminating Simulations:** For systems with a clear end (e.g., a bank open 9am-5pm). We analyze by running **multiple independent replications** and averaging the results.
- **Steady-State Simulations:** For continuous systems (e.g., a 24/7 server). We use techniques like **warm-up periods** (deleting initial data) to remove startup bias before collecting statistics.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF BARISAL
Final EXAMINATION 2021
Course Title: Simulation and Modeling
Course Code: CSE-3209
3^{rd} year 2^{nd} Semester
Session: 2020-21 (Admission: 2018-19)

Time: 3 hour **Marks:** 60

Answer any five Questions from the followings.

1. a) How simulation plays important role in new innovation? [3]
   b) In which situations simulation is not a better choice? [3]
   c) What are different methods of studying a system before designing a simulation? [3]
   d) Classify simulation models. [3]

2. a) Briefly explain the differences between discrete system and continuous system with suitable figures. [4]
   b) Discuss the concept of "Time Advance Mechanism" with an example. [4]
   c) What are different components of a Discrete-event simulation models? Explain. [4]

3. a) Define Chi – Square Test. [2]
   b) Generate three Poisson variants with mean $\alpha = 0.2$ using acceptance rejection techniques. Given random numbers are 0.4357, 0.4146, 0.8353, 0.9952 and 0.8004. [4]
   c) The life of a device used to inspect cracks in aircraft wings is given by X, a continuous random variable assuming all values in the range $x \geq 0$. The cdf of the device's lifetime, in years, is as follows- [6]

$$F(x) = \frac{1}{2}\int_0^x e^{-t/2}dt$$

   a. Find the probability that the device will last for <2 years.
   b. Find the probability that the device will last between 2 and 3 years.

4. a) Explain over all aspects and characteristics of a Queueing System. [8]
   b) Customers at a restaurant arrive in groups (one to eight persons). The number of persons (per group) for 300 customers and the relative frequencies appear are shown below. Draw empirical CDF. [4]

Arrivals per party	1	2	3	4	5	6	7	8
Frequency	30	110	45	71	12	13	7	12

5. a) Explain different methods of random number generation. [9]
   b) What are the challenges in generating pseudo random numbers. [3]

6. a) When Linear Congruential Generator is used? [2]