NIRMA UNIVERSITY

Institute:	Institute of Technology
Name of Programme:	BTech (CSE)
Course Code:	3CS207ME24
Course Title:	Quantum Computing
Course Type:	Disciplinary Minor-Elective
Year of Introduction:	2024-25

L	T	Practical Component				C
		LPW	PW	W	S	
3	0	2	-	-	-	4

Course Learning Outcomes (CLO):

At the end of the course, the students will be able to –

1. relate the differences between the classical and quantum systems (BL2)
2. apply the concepts of complex numbers to quantum computing (BL3)
3. examine the various algorithms using quantum programming (BL4)
4. develop various quantum applications. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	**Complex Numbers:** Basics, Algebra of Complex Numbers, Geometry of Complex Numbers, Properties, Examples, Basis, Dimension, Inner Products and Hilbert Spaces, Eigenvalues and Eigenvectors, Hermitian and Unitary Matrices, Tensor Product of Vector Spaces	06
Unit-II	**Leap from Classical to Quantum:** Classical Deterministic Systems, Probabilistic Systems, Quantum Systems, Assembling Systems	03
Unit-III	**Basic Quantum Theory:** Quantum States, Observables, Measuring, Dynamics, Assembling Quantum Systems, Bits and Qubits, Classical Gates, Reversible Gates, Quantum Gates	06
Unit-IV	**Algorithms:** Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Simon's Periodicity Algorithm, Grover's Search Algorithm, Shor's Factoring Algorithm	06
Unit-V	**Programming Languages:** Quantum Assembly Programming, Toward Higher-Level Quantum Programming, Quantum Computation Before Quantum Computers	06
Unit-VI	**Theoretical Computer Science:** Deterministic and Nondeterministic Computations, Probabilistic Computations, Quantum Computations	04
Unit-VII	**Cryptography:** Quantum Key Exchange Algorithms – BB84, B92 and EPR Protocols, Quantum Teleportation	04
Unit-VIII	**Information Theory:** Classical Information and Shannon Entropy, Quantum Information and von Neumann Entropy, Classical and Quantum Data Compression, Error Handling Codes	05
Unit-IX	**Hardware:** Goals and Challenges, Quantum Computers - Ion Traps, Linear Optics, NMR and Superconductors	05

Self-Study:
The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study contents

Suggested Readings/ References:

1. Noson S Yanofsky and Mirco A. Mannucci, Quantum Computing for Computer Scientists, Cambridge University Press
2. Vishal Sahni, Quantum Computing, Tata McGraw-Hill
3. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press
4. Robert S. Sutor, Dancing with Qubits: How quantum computing works and how it can change the world, Packt Publishing Limited
5. Eric Johnston, Nic Harrigan, Mercedes Gimeno–segovia, Programming Quantum Computers: Essential Algorithms and Code Samples, O'Reilly

Suggested List of Experiments:

Sr. No.	Title	Hours
1	• Write a program that accepts two complex numbers and outputs their sum and their product. • Improve the program to perform subtraction and division. Also, let the user enter a complex number and have the computer return its modulus and conjugate. • Write a program to convert a complex number from cartesian to polar and vice versa. • Add functions for multiplication, division, and returning the polar coordinates of a number.	02
2	• Write 3 functions to perform addition, inverse and scalar multiplication in complex vector spaces and matrices. • Write a function that accepts 2 complex vectors of length n and calculates their inner product, norm, and distance between two vectors. • Write a program to check whether a given square matrix is Hermitian or not, Unitary or not. • Write a program that accepts two matrices and constructs their tensor product.	04
3	• Write a program to perform the little marble experiment. The program should allow the user to enter a Boolean matrix that describes the ways those marbles move. The user should also be permitted to enter a starting state of how many marbles are on each vertex. Then, the user enters how many times she wants to proceed. The computer should then calculate and output the state of the system after those time clicks. • Modify the program to accept entries as fractions instead of Boolean values. • Modify the program to accept complex numbers instead of fractions.	04
4	• Write a program that simulates the first quantum system. The user should be able to specify how many points the particle can occur. The user will also specify a ket state vector by assigning its amplitudes. The program, when asked about the likelihood of finding the particle at a given point, will perform the calculations. If the user enters two kets,	04

the system will calculate the probability of transitioning from the first ket to the second after an observation has been made.

- Add observables to the picture: the user will input a square matrix of the appropriate size and a ket vector. The program will verify that the matrix is Hermitian and, if so, calculates the mean value and variance of the observable on the given state.
- When the user enters an observable and a state vector, the program will return the list of eigenvalues of the observable, the mean value of the observable on the state, and the probability that the state will transition to each one of the eigenstates.
- Apply dynamics to implementation.

5	Design and implement a quantum computer emulator in the language of your choice.	04
6	• Write functions that mimic Alice, Bob, and their interactions. Alice should generate two random bit strings of the same length. Call one BitSent and the other Sending Basis. Bob will be a function that generates a random bit string of the same length called ReceivingBasis. All three of these bit strings should be sent to an "all-knowing" function named Knuth. This function must look at all three and create a fourth-bit string named BitReceived. Knuth must furthermore evaluate the percentage of bits that Bob receives accurately. • Write a function named Knuth2 that accepts all three-bit strings and creates a bit string of possibly shorter length called AgreedBits, which is a substring of both BitSent and BitReceived.	04
7	Write 3 functions that mimic Alice, Bob, and their interactions. They should create bit strings that perform the B92 Protocol.	02
8	Write a program that lets the user choose how many letters the source alphabet has and then enter the probability distribution. The program should visualize it and compute its Shannon entropy.	02
9	Write a program that lets the user enter two qubits and their corresponding probabilities. Then calculate the density matrix, diagonalize it, and store the corresponding eigenbasis. The user will then enter a quantum message. The program will write the message in the eigenbasis and return the truncated part belonging to the typical subspace.	04