KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON

School of Computer Sciences

Bachelor of Science in Computer Science

Four Year B. Sc. (Computer Science)

Syllabus

[*As per NEP-2020*]

Faculty of Science and Technology

With effect from 2023-24

BACHELOR OF SCIENCE in Computer Science
[4-Year B.Sc. (Computer Science) as Per NEP-2020]

PROGRAMME OBJECTIVES (POs):

1) **Broadly Educated and Versatile** - Able to draw upon foundational knowledge, learn, adapt and successfully bring to bear analytical and computational approaches on changing societal and technological challenges.
2) **Inspiring and Collaborative** - Able to induce and contribute to diverse teams, expertise, and experiences.
3) **Innovative** - Drives scientific and societal advancement through technological innovation and entrepreneurship.
4) **Engaged** - Is and remains engaged with the academics, technical and scientific professional communities.

PROGRAMME SPECIFIC OUTCOMES (PSOs):

At the end of the program the graduate will be able to:

PSO No.	PSO	Cognitive level
BCSPSO.1	Apply computer science foundations, algorithmic principles, and computer science theory in the modeling and design of computational systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.	3
BCSPSO.2	Analyze a complex computing problem and apply principles of computing to identify solutions.	4
BCSPSO.3	Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	5
BCSPSO.4	Acquire and apply new knowledge as needed, using appropriate learning strategies.	3

KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON

School of Computer Sciences

Proposed Structure for B.Sc. (Computer Science) Level 4.5-6 Multidisciplinary Programme (2023-24)

[4 Year UG with SSC as Major]

Credit distribution structure for three/four year Honours/Honours with Research Degree Programme with Multiple Entry and Exit options

VSC-Vocational Skill Courses SEC-Skill Enhancement Courses AEC-Ability Enhancement Courses VEC-Value Education Courses

Level	Sem	Major		Minor (Choose Any One)		OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC, RP	Cum.Cr./ Sem.	Degree/Cum.Cr.
		Mandatory	Electives	Mathematics	Digital Electronics						
4.5	I	BCS-101: Programming Fundamentals & C++ Programming (2) (T) BCS-102: Computer Organization (2) (T) BCS-103: Lab on C++ Programming (2) (P)	-	-	-	OE-1: Choose from the OE Basket (4) (T)	BCS-104: Office Automation Tools (Hands-on) (2) BCS-105: Computer Hardware and Software Maintenance (Hands-on) (2)	AEC-1: Choose from the Basket (2) VEC-1: Choose from the Basket. (2) IKS-1: Choose from the Basket. (2)	CC-1: Choose from the Basket (2)	**22**	**UG Certificate44**
	II	BCS-106: Discrete Mathematics (2) (T) BCS-201: Object Oriented Programming with C++ (2) (T) BCS-202: Lab on Object Oriented Programming with C++ (2) (P)	-	MTM-101: Calculus (2) (T)	ETM-101: Digital Electronics (2) (T)	OE-2: Choose from the OE Basket (4) (T)	BCS-107: Network Administration and Maintenance (2) BCS-108: Word Processing Tools (Hands-on) (2) (T)	AEC-2: Choose from the Basket. (2) VEC-2: Choose from the Basket. (2)	CC-2: Choose from the Basket. (2)	**22**	
	Cum.Cr	**12**	**---**	**2**		**8**	**8**	**10**	**4**	**44**	
Exit option: Award of UG Certificate in Major with40-44 credits and an additional 4 credit score NSQF course/Internship OR Continue with Major and Minor											

Level	Sem	Major		Minor (Choose Any One)		OE	VSC, SEC (VSEC)	AEC, VEC,IKS	OJT, FP,CEP, CC,RP	Cum.Cr./ Sem.	Degree/Cum.Cr.
		Mandatory	**Electives**	**Mathematics**	**Digital Electronics**						
5.0	**III**	BCS-203: Operating System (2)(T) BCS-204: Data Structures(T)(2) BCS-205: Lab on Data Structures (2) (P) BCS-206: Lab on Linux administration (2)(P) 6(4+2)-8(2*4)		MTM-102: Graph Theory (4) (T)	ETM-102: Microprocessor Basics and Programming (4) (T)	OE-3:Choose from the OE Basket (T)(2)	BCS-207: Web Design-I (HTML, CSS, JavaScript) (Hands-on) (2)	AEC-3: Choose from the Basket. (2)	BCS-401: FP (2) CC-3: Choose from the Basket. (2)	**22**	**UG Diploma 88**
	IV	BCS-208: DBMS (2) (T) BCS-209: Java Programming (2) (T) BCS-210: Lab on Java Programming (2)(P) BCS-211: Lab on DBMS (2)(P) 6(4+2)-8(2*4)		MTM-103: Mathematics for Machine Learning (4)(T)	ETM-201: Microcontroller (4) (T)	OE-4:Choose from the OE Basket (T) (2)	BCS-212: Data Engineering (EDB /MongoDB) (Hands-on) (2)	AEC-4 Choose from the Basket. (2)	CC-4: Choose from the Basket. (2)	**22**	
	Cum .Cr	**28**	**---**	**10**		**12**	**12**	**14**	**12**	**88**	
Exit option; Award of UG Diploma in Major and Minor with 80-88 credits and an additional 4 credits core NSQF course/Internship OR Continue with Major and Minor											

Level	Sem	Major		Minor (Choose Any One)		OE	VSC, SEC (VSEC)	AEC, VEC,IKS	OJT, FP,CEP, CC,RP	Cum.Cr./ Sem.	Degree/Cum.Cr.
		Mandatory	**Electives**	**Mathematics**	**Digital Electronics**						
5.5	V **(DS)**	BDS-301: DAA (2)(T) BDS 302: Software Development Methodology (2)(T) BDS 303: Artificial Intelligence T (2) BDS-304: Lab on DAA (2)(P)	BCS-212 Introduction to Big Data Analytics (Swayam) (noc23-cs112)(T) (4) Or BCS-213: Computer Graphics (Swayam) (noc23-cs115) (T)(4)	MTM-104: Linear Algebra (T)(4)	ETM-202: IOT (T) (4)	-	BCS-301: Web Design-II (PHP, Angular, jQuery, bootstrap) ((2) BCS-302: Lab on Web Design-II (2)		BCS-402: FP-2(2)	**22**	**UG Degree 132**
	V **(IT)**	BIT-301: Mobile Application Development (2)(T) BIT-302: Artificial Intelligence (2)(T) BIT-303: Software Engineering(T)(2) BIT-304: Lab on MAD (2) (P)									
	VI **(DS)**	BDS-305: Machine Learning (2) (T) BDS-306: Data Visualization (2)(T) BDS-307: Python Programming (2)(T) BDS-308: Lab on Python Programming (2)(P) BDS-309: Lab on Data Visualization (2) (P)	BCS-303: Data Analytics with Python (Swayam) (noc23-cs08) T(4) Or BCS-304: NLP (Swayam) (noc23-cs45) (T)(4)	MTM-201: Theory of Computation (4)(T)	ETM-203 Digital Image Processing (4)(T)	-	-	-	BCS-403: OJT-1: (4)	**22**	
	VI	BIT-305: Machine									

	(IT)	Learning T(2) BIT-306: Web Design-III (Node.js, React)(T) (2) BIT-307: Python Programming (2)(T) BIT-308: Lab on Python Programming (2) (P) BIT-309: Lab on Web Design -III (2)(P)	BCS-305: Ethical Hacking (Swayam) (noc-23-cs75) T(4) Or BCS-306:Secure Computations: Part 1(Swayam) T(4)								
	Cum.Cr.	**48**	**08**	**20**		**10**	**14**	**14**	**18**	**132**	
		Exit option; Award of UG Diploma in Major and Minor with 80-88 credits and an additional 4 credits core NSQF course/Internship OR Continue with Major and Minor									

Level	**Sem**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC,IKS**	**OJT, FP,CEP, CC,RP**	**Cum.Cr./ Sem.**	**Degree/Cum.Cr.**
		Mandatory	**Electives**							
6.0	VII **(DS)**	BDS-310: Advanced Java(2)(T) BDS-311: Excel for Data Science(4)(T) BDS-312: SQL for Data Science (2)(T) BDS-313: Data Analytics (2)(T) BDS-314: Lab on Advanced Java (2)(P) BDS-315: Lab on SQL for Data Science (2)(P)	BCS-307: Cyber Security and Privacy T(4) (Swayam) noc23_cs127 Or BCS-308: Data Warehousing and Data Mining (DWDM)(4)(T)	RM (4)	-	-	-	-	**22**	**UG Honours** **176**
	VII **(IT)**	BIT-310: Computer Network(2) (T) BIT-311: Advanced Java (T) (4) BIT-312: Current Computing								

		Trends -I (T)(4) BIT-313:Lab on Advanced DBMS (No SQL) (2)(P) BIT-314: Lab on Advanced Java (2) (P)								
	VIII **(DS)**	BDS-316: Introduction to Neural Network with Deep Learning(4)(T) BDS-317: Cloud Concepts (Deploying Software for cloud implementation (4)(T) BDS-318: Web Technology (2)(T) BDS-319: Lab on Cloud Concepts (Deploying Software for cloud implementation (2)(P) BDS-320: Lab on WT (2)(P)	BCS-309: Compiler Design (T)(4) Or BCS-310: Blockchain Technology T (4)	-	-	-	-	BCS-404: OJT : (4)	**22**	
	VIII **(IT)**	BIT-316: MULTIMEDIA AND AR/VR (T)(4) BIT-317:Cloud Concepts (Deploying Software for cloud implementation) (4)(T) BIT-318: Advanced Software Development Methodologies (2)(T) BIT-319: Lab on Cloud Concepts (Deploying Software for cloud implementation) (2)(P) BIT-320: Lab on ASDM (2)(P)								
	Cum.Cr.	**76**	**16**	**24**	**10**	**14**	**14**	**22**	**176**	
Four Year UG Honours Degree in Major and Minor with 160-176 credits										

Level	Sem	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC,IKS	OJT, FP,CEP, CC,RP	Cum.Cr./ Sem.	Degree/Cum.Cr.
		Mandatory	**Electives**							
(6.0)	VII **(DS)**	BDS-321: Data Science and Analytics (2)(T) BDS-322: Advanced Software Development Methodologies(2) BDS-323: Introduction to Neural Network with Deep Learning (4)(T) BDS-324: Lab on Advanced Software Development Methodologies (2)(P)	BCS-311: Ruby on Rails T(2) BCS-312: Lab on Ruby on Rails (2)(P) Or BCS-313: Mobile Application Development T(2) BCS-314: Lab on MAD (2)(P)	RM (4)	-	-		BCS-405: RP:4	**22**	**UG Honours with Research Degree 176**
	VII **(IT)**	BIT-321: Advanced Software Development Methodologies(2)(T) BIT-322: Front End Programming (PHP/ Python)T(2) BIT-323: Current Computing Trends -I (T) (2) BIT-324: Lab on Front End Programming (PHP/ Python) (2) (P) BIT-325: Lab on Advanced Software Development Methodologies (2)(P)			-	-				
	VIII **(DS)**	BDS-325: Front End Framework (Angular, React, Node.js) (2)(T) BDS-326:Cloud Concepts (Deploying Software for cloud implementation) (2)(T) BDS-327: Digital Image Processing & Computer Vision (2)(T) BDS-328: Lab on Cloud Concepts (Deploying Software for cloud implementation) (2)(P)	BCS-315: Compiler Design T(4) Or BCS-316: Blockchain Technology T(4)	- -	-	-	-	BCS-406 :RP: 8	**22**	

		BDS-329: Lab on Front End Framework (Angular, React, Node.js) (2)(P)								
	VIII **(IT)**	BIT-326: Cloud Concepts (Deploying Software for cloud implementation) (2)(T) BIT.327: Front End Framework (Angular, React, Node.js) (2)(T) BIT-328:MULTIMEDIA AND AR/VR (T)(2) BIT-329: Lab on Cloud Concepts (Deploying Software for cloud implementation) (2)(P) BIT-330: Lab on Front End Framework (Angular, React, Node.js)(2)(P)			-	-	-			
	Cum.Cr.	**68**	**16**	**24**	**10**	**14**	**14**	**30**	**176**	
FourYearUGHonourswithResearchDegreeinMajorandMinorwith160-176 credits										

School of Computer Sciences

B.Sc.(Level4.5)Sem-I(Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course(Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BCS-101	Programming Fundamentals & C++ Programming	2	30	2	-	8	20	12	30	2
		BCS-102	Computer Organization	2	30	2	-	8	20	12	30	2
		BCS-103	Lab on C++ Programming	2	60	-	4	8	20	12	30	2
2	OE (Open Elective) **(Any One)**	OEMG-101	Financial Accountancy	4	60	4	-	16	40	24	60	3
		OESS-101	Leadership Development									
		OEMS-101	Statistical Methods									
3	VSC/SEC (VSEC) Skill Courses	BCS-104	Office Automation Tools (Hands-on)	2	30	2	-	8	20	12	30	2
		BCS-105	Computer Hardware and Software Maintenance (Hands-on)	2	30	2	-	8	20	12	30	2
4	AEC **(Any One)**	AEC-1	Select One from the AEC Basket	2	30	2	-	8	20	12	30	2
	VEC **(Any One)**	VEC-1	Select One from the VEC Basket	2	30	2	-	8	20	12	30	2
	IKS **(Any One)**	IKS-1	Select One from the IKS Basket	2	30	2	-	8	20	12	30	2
5	CC Co-curricular **(Any One)**	CC-1	Select One from the CC Basket	2	30	2	-	8	20	12	30	2
Total				22		20	4	220		330		

School of Computer Sciences

B.Sc.(Level4.5)Sem- II(Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BCS-106	Discrete Mathematics	2	30	2	-	8	20	12	30	2
		BCS-201	Object Oriented Programming with C++ (2) (T)	2	30	2	-	8	20	12	30	2
		BCS-202	Lab on Object Oriented Programming with C++ (2) (P)	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-101	Calculus	2	30	2	-	8	20	12	30	2
		ETM-101	Digital Electronics									
3	OE (Open Elective) **(Any One)**	OEMG-102	Corporate Communication	4	60	4	-	16	40	24	60	3
		OESS-102	Indian Civil Services									
		OEMS-102	Actuarial Methods									
4	VSC/SEC (VSEC) Skill Courses	BCS-107	Network Administration and Maintenance	2	30	2	-	8	20	12	30	2
		BCS-108	Word Processing Tools (Hands-on)	2	30	2	-	8	20	12	30	2
5	AEC **(Any One)**	AEC-2	Select One from the AEC Basket	2	30	2	-	8	20	12	30	2
	VEC **(Any One)**	VEC-2	Select One from the VEC Basket	2	30	2	-	8	20	12	30	2
6	CC Co-curricular **(Any One)**	CC-2	Select One from the CC Basket	2	30	2	-	8	20	12	30	2
Total				22		20	4	220		330		

School of Computer Sciences

B.Sc.(Level5)Sem- III(Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BCS-203	Operating System	2	30	2	-	8	20	12	30	2
		BCS-204	Data Structures (2) (T)	2	30	2	-	8	20	12	30	2
		BCS-205	Lab on Data Structures (2) (P)	2	60	-	4	8	20	12	30	2
		BCS-206	Lab on Linux administration	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-102	Graph Theory	4	60	4	-	16	40	24	60	3
		ETM-102	Microprocessor Basics and Programming									
3	OE (Open Elective) **(Any One)**	OEMG-201	Startup and Innovation	2	30	2	-	8	20	12	30	2
		OESS-201	Social Marketing									
		OEMS-201	Sampling Methods									
4	VSC/SEC (VSEC) Skill Courses	BCS-207	Web Design-I (HTML, CSS, JavaScript) (Hands-on)	2	30	2	-	8	20	12	30	2
5	AEC **(Any One)**	AEC-3	Select One from the AEC Basket	2	30	2	-	8	20	12	30	2
6	CC Co-curricular **(Any One)**	BCS-401	FP	2	30	2	-	8	20	12	30	2
		CC-3	Choose from the Basket									
			Total	20		16	08		200		300	

School of Computer Sciences

B.Sc.(Level5)Sem- IV(Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BCS-208	Data Base Management System	2	30	2	-	8	20	12	30	2
		BCS-209	Java Programming	2	30	2	-	8	20	12	30	2
		BCS-210	Lab on Java Programming	2	60	-	4	8	20	12	30	2
		BCS-211	Lab on DBMS	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-103	Mathematics for Machine Learning	4	60	4	-	16	40	24	60	3
		ETM-201	Microcontroller									
3	OE (Open Elective) **(Any One)**	OEMG-202	Entrepreneurial Competencies	2	30	2	-	8	20	12	30	2
		OESS-103	Industrial Sociology									
		OEMS-202	Testing of Hypothesis									
4	VSC/SEC (VSEC) Skill Courses	BCS-212	Data Engineering (EDB/MongoDB) (Hands-on)	2	30	2	-	8	20	12	30	2
5	AEC **(Any One)**	AEC-4	Select One from the AEC Basket	2	30	2	-	8	20	12	30	2
6	CC Co-curricular **(Any One)**	CC-4	Choose from the Basket	2	30	2	-	8	20	12	30	2
			Total	20		16	08		200		300	

School of Computer Sciences

B.Sc.(Level5.5)Sem- V(Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BDS-301	Design and Analysis of Algorithms	2	30	2	-	8	20	12	30	2
		BDS 302	Software Development Methodology	2	30	2	-	8	20	12	30	2
		BDS 303	Artificial Intelligence	2	30	2	-	8	20	12	30	2
		BDS-304	Lab on DAA	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-104	Linear Algebra	4	60	4	-	16	40	24	60	3
		ETM-202	IOT									
3	Major (Elective) **(Any One)**	BCS-212	Introduction to Big Data Analytics(Swayam)	4	60	4	-	16	40	24	60	3
		BCS-213	Computer Graphics(Swayam)									
4	VSC/SEC (VSEC) Skill Courses	BCS-301	Web Design-II (PHP, Angular, jQuery bootstrap)	2	30	2	-	8	20	12	30	2
		BCS-302	Lab on Web Design-II	2	60	-	4	8	20	12	30	2
5	CC Co-curricular **(Any One)**	BCS-402	FP-2	2	30	2	-	8	20	12	30	2
			Total	22		18	8		220		330	

School of Computer Sciences

B.Sc. IT (Level 5.5)Sem- V(Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BIT-301	Mobile Application Development	2	30	2	-	8	20	12	30	2
		BIT-302	Artificial Intelligence	2	30	2	-	8	20	12	30	2
		BIT-303	Software Engineering	2	30	2	-	8	20	12	30	2
		BIT-304	Lab on MAD	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-104	Linear Algebra	4	60	4	-	16	40	24	60	3
		ETM-202	IOT									
3	Major (Elective) **(Any One)**	BCS-212	Introduction to Big Data Analytics (Swayam)	4	60	4	-	16	40	24	60	3
		BCS-213	Computer Graphics (Swayam)									
4	VSC/SEC (VSEC) Skill Courses	BCS-301	Web Design-II (PHP, Angular, jQuery bootstrap)	2	30	2	-	8	20	12	30	2
		BCS-302	Lab on Web Design-II	2	60	-	4	8	20	12	30	2
5	CC Co-curricular **(Any One)**	BCS-402	FP-2	2	30	2	-	8	20	12	30	2
			Total	22		18	8		220		330	

School of Computer Sciences

B.Sc.(Level 5.5)Sem- VI (Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BDS-305	Machine Learning	2	30	2	-	8	20	12	30	2
		BDS-306	Data Visualization	2	30	2	-	8	20	12	30	2
		BDS-307	Python Programming	2	30	2	-	8	20	12	30	2
		BDS-308	Lab on Python Programming	2	60	-	4	8	20	12	30	2
		BDS-309	Lab on Data Visualization	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-201	Theory of Computation	4	60	4	-	16	40	24	60	3
		ETM-203	Digital Image Processing									
3	Major (Elective) **(Any One)**	BCS-303	Data Analytics with Python (Swayam)	4	60	4	-	16	40	24	60	3
		BCS-304	NLP(Swayam)									
4	CC Co-curricular **(Any One)**	BCS-403	OJT-1	4	60	4	-	16	40	24	60	4
			Total	22		18	8		220		330	

School of Computer Sciences

B.Sc. IT(Level 5.5)Sem- VI (Name of Courses for -Major, Minor, OE, VSEC, IKS courses)

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/Semester	Teaching Scheme(hrs/week)		Evaluation Scheme				
						Theory	Practical	Continuous Internal Evaluation (CIE)(CA)		End Semester Evaluation (ESE)(UA)		Duration of Exam.(Hrs)
						T	P	Min	Max	Min	Max	
1	SSC (Major) Mandatory	BIT-305	Machine Learning	2	30	2	-	8	20	12	30	2
		BIT-306	Web Design-III	2	30	2	-	8	20	12	30	2
		BIT-307	Python Programming	2	30	2	-	8	20	12	30	2
		BIT-308	Lab on Python Programming	2	60	-	4	8	20	12	30	2
		BIT-309	Lab on Web Design -III	2	60	-	4	8	20	12	30	2
2	Minor (Mathematics) **OR** Minor (Digital Electronics)	MTM-201	Theory of Computation	4	60	4	-	16	40	24	60	3
		ETM-203	Digital Image Processing									
3	Major (Elective) **(Any One)**	BCS-305	Ethical Hacking (Swayam)	4	60	4	-	16	40	24	60	3
		BCS -306	Secure Computations: Part 1 (Swayam)									
4	CC Co-curricular **(Any One)**	BCS-403	OJT-1	4	60	4	-	16	40	24	60	4
			Total	22		18	8		220		330	

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon

UG Honours Programs Minor A.Y. 2023-24

Sem	Code	Minor-Economics	Code	Minor-Data Science	Credit
II	ECM-101	Indian Economy	DSM-101	Introduction to Data Science	02
III	ECM-102	Micro-economics	DSM-102	Data Analysis with Python	04
IV	ECM-103	Macro-Economics	DSM-103	Database Management System	04
V	ECM-201	Financial Economics	DSM-201	Machine Learning for Data Science	04
VI	ECM-202	Money, Banking and Public Finance	DSM-202	Artificial Intelligence	04

Sem	Code	Minor-Statistics	Code	Minor-Mathematics	Credit
II	STM-101	Basic Probability	MTM-101	Calculus	02
III	STM-102	Descriptive Statistics	MTM-102	Graph Theory	04
IV	STM-103	Probability Distributions	MTM-103	Mathematics for Machine Learning	04
V	STM-201	Sample Surveys and Official Statistics	MTM-104	Linear Algebra	04
VI	STM-202	Statistical Inference	MTM-201	Theory of Computation	04

Sem	Code	Minor-Digital Electronics	Code	Minor-Management	Credit
II	ETM-101	Digital Electronics	MGM-101	Principles of Management	02
III	ETM-102	Microprocessor Basics and Programming	MGM-201	Strategic Management	04
IV	ETM-201	Microcontrollers	MGM-102	Human Resource Management	04
V	ETM-103	Internet of Things (IOT)	MGM-103	Marketing Management	04
VI	ETM-202	Digital Image Processing	MGM-202	Finance and Investment Management	04

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon

UG Programs Open Elective (OE) Baskets A.Y. 2023-24

Semester (Credit)	Computer Science	Management	Social Sciences	Mathematical Sciences (Statistics)
I (4)	OECS-101: Spreadsheets Tools (Hands-on) (T)(4)	OEMG:101: Financial Accountancy (4)	OESS-101: Leadership Development (4)	OEMS-101: Statistical Methods (4)
II (4)	OECS-102: Word Processing Tools (Hands-on) (T)(4)	OEMG-102: Corporate Communication (4)	OESS-102: Indian Civil Services (4)	OEMS-102: Actuarial Methods (4)
III (2)	OECS-201: Web Design-I (2)	OEMG-201: Startup and Innovation (T)(2)	OESS-201: Social Marketing (2)	OEMS-201: Sampling Methods (2)
IV (2)	OECS-202 Data Base Management Systems (DBMS) (T) (2) (Select only one DCM-103 or OECS-202)	OEMG-202: Entrepreneurial Competencies (T) (2)	OESS-103: Industrial Sociology (2)	OEMS-202: Testing of Hypothesis (2)

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon

UG Programs Vertical 5 and 6 Baskets A.Y. 2023-24

AEC (2) (Any One)	VEC (2) (Any One)	IKS (2) (Any One)	CC, CEP (2) (Any One)
AEC-101)Communicative English-I AEC-102) Communicative English-II AEC-103) Employability Skills AEC-104) Design Thinking	VEC-101) Environmental Awareness VEC-102) Fundamentals of Indian Constitution	IKS-101) An Ancient Indian Science of Exercise and Healing IKS-102) Introduction to Indian Traditional Performing Arts IKS-103) Vedic Mathematics	CC-101) National Service Scheme (NSS) CC-102) Practicing Sports CC-103) Music CC-104) Integrated Personal Development Course (IPDC)

Detail Syllabus
(Semester I and II)

Semester-I

BCS-101: Programming Fundamentals & C++ Programming
*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To familiarize students to C++ programming language, fundamentals of programming.
- To understand the basic principles of programming using C++, algorithmic and procedural problem solving, program design and development, basic data types, control structures, functions, arrays, pointers, and introduction to classes for programmer-defined data types.
- To understand and know the importance of pointers.

Unit-I: Introduction **L: 05**

Introduction, Basic block diagram and functions of various components of computer, Concept of Hardware and Software, Types of software, Compiler and Interpreter. Basic Difference between Procedure Oriented Language and Object-Oriented Language, Concepts of Machine level, Assembly level and High-level programming, Flow charts and Algorithm.

Unit-II: Introduction to C++ **L: 05**

Introduction to C++, Features of C++ language, Structure of a C++ program, character set, keywords and identifiers, data types, variable declaration, reference variables, operators in C++, input and output statements, symbolic constant definition, input and output statements, type compatibility.

Unit III: Operators and Expressions **L: 05**

C++ operators- arithmetic, relational, logical, bitwise, assignment, increment and decrement, conditional (?:) and special operators, Arithmetic expressions, precedence of operators and associatively. Type conversions, mathematical functions. Definition of macro and pre-processor directives, Managing I/O operation – reading and writing a character.

Unit IV: Control Structures **L: 05**

Conditional control statements- if, if-else, nested if, switch, go to statement, while, do-while and for statements. Unconditional control statements- break, continue and return statements (definition and explanation with syntax, flowchart and examples)

Unit V: Arrays, Strings, structure, Unions **L: 05**

Definitions of an array, types of arrays, (definition, declaration, initialization with examples). **Strings**: definition, declaration and initialization of string variable, string handling functions (explanation with syntax and examples) structure declaration and definition, use of structure and union, difference between structure and unions.

Unit VI: Function, Recursion and Pointer **L: 05**

Function component, parameter passing – pass by value, pass by address, pass by reference, inline function, function overloading, scope and extent of variables, **Recursion:** Definitions, recursive function, Examples, Applications, recursive function pointer variables, address operator & Runtime memory management, pointer to pointer, array of pointer, pointer constant, pointer arithmetic, pointer to function, pointer to objects, array of object, this pointer, self-referential classes.

References:

1. V. Rajaraman, Neeharika Adabala, Fundamentals of Computers, sixth edition, ISBN: 978-81-203-5067-0,
2. Robert Lafore, Object-Oriented Programming in C++, fourth edition, Sams Publishing, ISBN:0672323087.
3. Bjarne Stroustrup, The C++ Programming language, Third edition, Pearson Education ISBN 0-201-88954-4.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS101C.1	Understand and use the basic programming constructs of C++ and manipulate various C++ datatypes, such as arrays, strings, and pointers.	2
BCS101C.2	Develop allocation/deallocation procedures for memory management.	6
BCS101C.3	Design/Write small-scale C++ programs using the above skills.	6

BCS-102: Computer Organization

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Discuss the basic concepts and structure of computers.
- Understand concepts of register transfer logic and arithmetic operations.
- Explain different types of addressing modes and memory organization.
- Learn the different types of serial communication techniques.
- Summarize the Instruction execution stages.

Unit-I:Data Representation **L: 06**

Number System, Binary, Octal, Decimal, Hexadecimal and BCD, 1's & 2's complement, arithmetic manipulation, addition, subtraction, Multiplication, Division, Fixed point representation, Floating point representation.

Unit-II: Digital Logic Circuit **L: 06**

Logic gets, AND, OR, NOT, NOR, XOR, NAND, Boolean Algebra map simplification, Combinational circuit, sequential circuit, flip flop.

Unit-III: Principal of Computer Design **L: 08**

Software, Hardware layers in computer system, Machine language instruction, Addressing modes, and types of instruction. Instruction Cycle and execution cycle, Micro programming Vs hardwired control, RISC Vs CISC, Superscalar processor, Arithmetic and logic unit, Introduction to array processor and its application, Central processing unit, Interrupt control unit.

Unit-IV: Memory Organization **L: 05**

Memory system, storage Technologies, Memory array organization, Memory hierarchy, Memory interleaving, cache and virtual memories.

Unit-V: I/O Devices **L: 05**

Input output devices and characteristics, Input-Output processing, Bus interface, Data transfer techniques, I/O Interrupt, Channels.

References:

1. M. Morris Mano, Computer System Architecture, Prentice Hall of India Pvt. Ltd., Eastern Economy Edition
2. Ramesh S Goankar, Microprocessor Architecture, Programming & Applications with the 8085, Penram International Publishing (India) Pvt. Ltd.
3. V. Heuring, H. Jorden, T. Venkatesh, Computer Systems Design and Architecture, Pearson Education, Second Ed., 2009.
4. William Stallings, Computer Architecture and Organization, PHI Pvt. Ltd., Eastern Economy Edition.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS102C.1	Identify, understand, and **apply** different number systems and codes.	2
BCS102C.2	Understand the digital representation of data in a computer system. Also, general concepts in digital logic design, including logic elements, and their **use in** combinational and sequential logic circuit design.	4
BCS102C.3	Understand computer arithmetic formulate and **solve problems**, understand the performance requirements of systems.	3

BCS-103: Lab on C++ Programming

Course Objectives:

The objectives of the course are-

- Learn the syntax and semantics of the C++ programming language.
- Apply Tokens, expressions and control structures in C++.
- Arranging same data systematically with arrays in C++.

---

1. Write a CPP Program to understand the basic data types and I/O.
2. Write a CPP Program on Operators and Expressions
3. Write a CPP Program on decision statements.
4. Write a CPP Program on looping.
5. Write a CPP Program on arrays.
6. Write a CPP Program on functions.

7. Write a CPP Program on structures and unions.
8. Write a CPP Program on pointers.
9. Write a CPP Program on string manipulations.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS103C.1	Develop logic of a program for solving real time problems and isolate and fix common errors in C++ programs.	5

BCS-104: Office Automation Tools (Hands-on)

ClockHours: **60**

Course Objective:

The objectives of the course are-

- The course would enable the students to craft professional DTP, word documents, excel spread sheets, power point presentations using the Microsoft suite of office tools.
- To familiarize the students in preparation of documents and presentations with office automation tool.

Unit-I: Desktop Publishing **L: 10**

MS-Windows: Introduction to Computer, Computer Basic, Creating Folder, Directories. **PageMaker**: Type Settings for Publication, Page Layout, Word Wrapping, Grouping, Merging two or more files, Creating columns, Tab settings, Paragraph settings, Hyphenation, Paper Style, Index & Table of Contents, Fonts, Mixing Text & Graphics, Linking objects, Printing facility. **Photoshop:** All Tools (Marquee Tool, Magnetic Tool, Slice Tool, Patch Tool, Clone Stamp Tool, Gradient Tool, Smudge Tool, Blur Tool, Text Tool etc.) Fill, Stroke Option Histogram, Group, Ungroup Lock Object, Color Range Feather, Modify, Grow, Filter Liquify, Artistic Blur, Video Option etc.

Unit-II: MS Word **L: 15**

Working with Documents -Opening & Saving files, Editing text documents, Inserting, Deleting, Cut, Copy, Paste, Undo, Redo, Find, Search, Replace, Formatting page & setting Margins, Converting files to different formats, Importing & Exporting documents, Sending files to others, Using Tool bars, Ruler, Using Icons, using help, Formatting Documents - Setting Font styles, Font selection- style, size, colour etc, Type face - Bold, Italic, Underline, Case settings, Highlighting, Special symbols, Setting Paragraph style, Alignments, Indents, Line Space, Margins, Bullets & Numbering. Setting Page style - Formatting Page, Page tab, Margins, Layout settings, Paper tray, Border & Shading, Columns, Header & footer, Setting Footnotes & end notes – Shortcut Keys; Inserting manual page break, Column break and line break, Creating sections & frames, Anchoring & Wrapping, Setting Document styles, Table of Contents, Index, Page Numbering, date & Time, Author etc., Creating Master Documents, Web page. Creating Tables- Table settings, Borders, Alignments, Insertion, deletion, Merging, Splitting, Sorting, and Formula, Drawing - Inserting ClipArts, Pictures/Files etc., Tools – Word Completion, Spell Checks, Mail merge, Templates, Creating contents for books, Creating Letter/Faxes, Creating Web pages, Using Wizards, Tracking Changes, Security, Digital Signature. Printing Documents – Shortcut keys.

Unit-III: MS Excel **L: 15**

Spread Sheet & its Applications, Opening Spreadsheet, Menus - main menu, Formula Editing, Formatting, Toolbars, Using Icons, Using help, Shortcuts, Spreadsheet types. Working with Spreadsheets- opening, Saving files, setting Margins, Converting files to different formats (importing, exporting, sending files to others), Spread sheet addressing - Rows, Columns & Cells, Referring Cells & Selecting Cells – Shortcut Keys. Entering & Deleting Data- Entering data, Cut, Copy, Paste, Undo, Redo, Filling Continuous rows, columns, Highlighting values, Find, Search & replace, Inserting Data, Insert Cells, Column, rows & sheets, Symbols, Data from external files, Frames, Clipart, Pictures, Files etc, Inserting Functions, Manual breaks, Setting Formula - finding total in a column or row, Mathematical operations (Addition, Subtraction, Multiplication, Division, Exponentiation), Using other Formulae. Formatting Spreadsheets- Labelling columns & rows, Formatting- Cell, row, column & Sheet, Category - Alignment, Font, Border & Shading, Hiding/ Locking Cells, Anchoring objects, Formatting layout for Graphics, Clipart etc., Worksheet Row & Column Headers, Sheet Name, Row height & Column width, Visibility - Row, Column, Sheet, Security, Sheet Formatting & style, Sheet background, Colour etc, Borders & Shading – Shortcut keys. Working with sheets – Sorting, Filtering, Validation, Consolidation, and Subtotal. Creating Charts - Drawing. Printing. Using Tools – Error checking, Spell Checks, Formula Auditing, Creating & Using Templates, Pivot Tables, Tracking Changes, Security, Customization.

Unit-IV: MS Power point **L: 10**

Opening new presentation, Different presentation templates, setting backgrounds, Selecting presentation layouts. Creating a presentation - Setting Presentation style, Adding text to the Presentation. Formatting a Presentation - Adding style, Colour, gradient fills, arranging objects, Adding Header & Footer, Slide Background, Slide layout. Adding Graphics to the Presentation- Inserting pictures, movies, tables etc into presentation, Drawing Pictures using Draw. Adding Effects to the Presentation- Setting Animation & transition effect. Printing Handouts, Generating Standalone Presentation viewer.

Unit-V: Internet & Advanced Communication **L: 10**

Internet and Web Browsers: Definition & History of Internet - Uses of Internet - Definition of Web Addressing-URL-Different types of Internet Connections; Dial up connection, Broad band (ISDN, DSL, Cable), Wireless (Wi-Fi, WiMax, Satellite, Mobile) naming convention, browsers and its types, internet browsing, searching - Search Engines - Portals - Social Networking sites- Blogs - viewing a webpage, downloading and uploading the website; Creating an email-ID, e-mail reading, saving, printing, forwarding and deleting the mails, checking the mails, viewing and running file attachments, addressing with cc and bcc. **Introduction to various devices & Applications:** Other than the computers, (electronic gadgets), which are widely using by executives in the Offices – Tablet, Smart Phone – concept of mobile phone and Tablet and their uses – Various applications using by Tablets and Smart Phones such as UC browser, WhatsApp, Maps, Skype.

Textbooks:

1. Vikas Gupta, Comdex Information Technology course tool kit, WILEY Dreamtech,2005.
2. Cheryl A Schmidt, The Complete Computer upgrade and repair book,3rd edition, WILEY Dreamtech.
3. Vikas Gupta, Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
4. Kate J. Chase, PC Hardware- A Handbook, PHI (Microsoft), ISBN Code: 978-0735620490.
5. Bittu Kumar, Mastering MS Office, V&S Publishers, 2017, ISBN, 9350578786, 9789350578780.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS104C.1	To perform/create documentation	6
BCS104C.2	To implement accounting operations	3
BCS104C.3	Develop presentations with PowerPoint software	6

BCS-105: Computer Hardware and Software Maintenance (Hands-on)

*ClockHours:***30**

Course Objective:

The objectives of the course are-

- Understand basic concept & structure of Computer Hardware & Networking Components.
- Identify the existing configuration of the computers & peripherals.
- Upgrading the same as & when required.
- Apply their knowledge about computer peripherals to identify/rectify problems on board.
- Integrate the PC's into Local Area Network & re-install OS & various shipboard applications.
- Perform routine maintenance, upgrades of virus definitions, set schedules etc.
- Manage data backup & restore operations.
- Analyze &/or troubleshoot communication problems related to Engine Room Automation & various Navigational Bridge Equipment.

Unit 1: Hardware Maintenance **L: 15**

- Introduction of computer hardware components
- Handling following hardware problems/issues:
  1) Introduction to computer components.
  2) Not Switching On CPU / Laptop
     a. Power issue
     b. Mains cable
     c. On off switch
     d. SMPS
     e. Motherboard
  3) No Booting
     a. Operating system corrupt
     b. Hard disk Issue
     c. Boot Sequence
     d. RAM issue
     e. CMOS issue
     f. No ejection of CD / DVD Drive
  4) Monitor not switching on (No display)
  5) Printer / Scanner / Mouse / keyboard (accessories) not working

6) Network card related problems
7) UPS Related problems
8) Battery related problems
9) Modem related problems
10) Use of multi meter
11) Installation of SSD on CPU / Laptop
12) Laptop adaptor and its issue
13) Laptop battery issue
14) Bluetooth devices hardware
15) Wi- Fi related problems
16) SERVERS special connectors

Unit 2: Software Maintenance **L: 15**

- Operating systems Window / Ubuntu Installation for Laptop Desktop.
  1) Operating systems Installation for SERVER
  2) SERVER Policy's
  3) Driver Installation.
  4) Various Software Installations.
  5) Disk Management.
  6) Device Manager.
  7) MS Office and its components'
  8) Virus and Antivirus
  10) Password issues
  11) Recovery software's
  12) Useful software's
      - Rufus (Boot disk preparation)
      - Hard disk cloning
      - DOS commands

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS105C.1	Identify, at a basic level, the names, purposes, and characteristics, of system modules, and recognize the modules by sight or definition.	2
BCS105C.2	Identify, at a basic level, the major desktop components and interfaces, and their functions, and differentiate the characteristics of the various desktop operating systems.	2
BCS105C.3	Identify, at a basic level, the procedures for adding and removing field-replaceable modules for desktop systems and portable systems and, given a replacement scenario, choose the appropriate sequences Demonstrate, at a basic level, the ability to use command line functions and utilities to manage the operating system, including proper syntax and switches.	3
BCS105C.4	Identify, at a basic level, the proper procedures for installing and configuring common SATA devices and peripheral devices and choose the appropriate installation or configuration sequences in given scenarios, as well as recognize the associated cables.	2
BCS105C.5	Identify, at a basic level, the steps to perform an operating system	3

	upgrade, and given an upgrade scenario, choose the appropriate next steps.	
BCS105C.6	Recognize, at a basic level, common problems associated with each module and their symptoms, identify steps to isolate and troubleshoot the problems, and given a problem situation, interpret the symptoms and infer the most likely cause.	5

Semester-II

BCS-106: Discrete Mathematics

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To familiarize concepts of discrete mathematics, such as set theory, relations, sequences, recursion, function, graphs, tree , counting and probability etc.
- To solve problems of relations, functions, logic of compound, counting, probability relations etc.
- To understand and know the importance of graphs, trees and probability.

Unit-I: Introduction **L: 05**

Variables, The Language of Sets, The Language of Relations and Function Set Theory: Definitions and the Element Method of Proof, Properties of Sets, Algebraic Proofs, Boolean Algebras, The Logic of Compound **Statements:** Logical Form and Logical Equivalence, Conditional Statements, Valid and Invalid Arguments.

Unit-II: Sequences and Recursion **L: 05**

Sequences, defining sequences recursively, solving recurrence relations by iteration.

Unit-III: Functions **L: 05**

Functions Defined on General Sets, One-to-One and Onto, Inverse Functions, Composition of Functions, Cardinality with Applications to Computability

Unit-IV: Relations **L: 04**

Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations

Unit-V: Graphs and Trees **L: 05**

Definitions and Basic Properties, Trails, Paths, and Circuits, Matrix Representations of Graphs, Isomorphism's of Graphs, Trees, Rooted Trees, Isomorphism's of Graphs, Spanning trees and shortest paths.

Unit-VI: Counting and Probability **L: 06**

Introduction, Possibility Trees and the Multiplication Rule, Possibility Trees and the Multiplication Rule, Counting Elements of Disjoint Sets: The Addition Rule, Conditional Probability, Bayes' Formula, and Independent Events.

Textbook:

1. Kenneth H. Rosen, Discrete Mathematics and Its Applications, Seventh Edition, McGraw Hill Education (India) Private Limited. (2011).
2. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.

References:

1. C. L Liu, Elements of Discrete Mathematics, McGraw-Hill Inc, 1985.

2. Alan Tucker, Applied Combinatorics 6th Edition, Wiley Publication,2012, ISBN: 978-0-470-45838-9
3. Ronald Graham, Donald Knuth, and Oren Patashnik, Concrete Mathematics, 2nd Edition - Pearson Education Publishers - 1996.
4. Peter J. Cameron, Combinatorics: Topics, Techniques, Algorithms, Cambridge University Press, 1994 (reprinted 1996).
5. I.N. Herstein, Topics in Algebra 2nd Edition, Wiley, 1975.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS201C.1	To understand an overview of the theory of discrete objects, starting with relations and partially ordered sets.	2
BCS201C.2	Make use of recurrence relations, generating function and operations on them.	3
BCS201C.3	Give an understanding of graphs and trees, which are widely used in software.	2
BCS201C.4	To understand basic knowledge about counting and probability.	2

BCS-201: Object Oriented Programming with C++

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To familiarize the Object-Oriented Programming (OOP) concepts, such as abstraction, encapsulation, instances, initializations, polymorphism, overloading, inheritance etc.
- To write programs to solve problems using generic programming constructs such as templates and using standard template library.
- To understand and know the importance of pointers and learn file handling and exception handling in real-world problems.

Unit I: Introduction **L: 06**

What is object-oriented programming? Why do we need object-oriented. Programming characteristics of object-oriented languages. **Introduction to Classes and Objects:** Classes in C++, class declaration, declaring objects, Defining Member functions, Inline member function, Array of objects, Objects as function argument, Static data member and member function, Friend function and friend class.

Unit II: Constructors and Destructors **L: 06**

Constructors, Instantiation of objects, Default constructor, Parameterized constructor, Copy constructor and its use, Destructors, Constraints on constructors and destructors, Dynamic initialization of objects.

Unit III: Polymorphism **L: 08**

Operator Overloading: concept of overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Keywords ‘explicit’ and ‘mutable’.

Virtual Functions: Virtual functions, pure virtual functions; Polymorphism, Categorization of polymorphism techniques: Compile time polymorphism, Run time polymorphism.

Unit IV: Inheritance **L: 04**

Derived class and base class: Defining a derived class, Accessing the base class member, Inheritance: multilevel, multiple, hierarchical, hybrid; Virtual base class, Abstract class.

Unit V: File Handling and Templates **L: 06**

File classes, Opening and Closing a file, File modes, Manipulation of file pointers, Functions for I/O operations. Templates: Function templates, Template specialization, Class templates, Non-type parameters for templates, template, and inheritance, The typename and export keywords.

References:

1. Robert Lafore, Object-Oriented Programming in C++, fourth edition, Sams Publishing, ISBN:0672323087.
2. Bjarne Stroustrup, The C++ Programming language, Third edition, Pearson Education ISBN 0-201-88954-4.
3. Meeta Gandhi, Tilak Shetty, Rajiv Shah, Vijay Mukhi's The 'C' Odyssey C++, and Graphics- The future of C, BPB publications, First Edition

---

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS202C.1	Understand and use the basic programming constructs of C++ and manipulate various C++ datatypes, such as arrays, strings, and pointers.	2
BCS202C.2	Manage memory appropriately using proper allocation/deallocation procedures.	3
BCS202C.3	Write small-scale C++ programs using the above skills.	6

BCS-202: Lab on Object Oriented Programming with C++

Course Objectives:

The objectives of the course are-

- Apply object-oriented approaches to software problems in C++.
- Apply exception handling techniques to software problems in C++.
- Apply generic programming approaches using templates and efficiently use standard template library in software development.

---

1. Write program to demonstrate class, use of constructor, constructor overloading and destructor.
2. Write program to demonstrate use of operator overloading.

3. Write program(s) to demonstrate use of inheritance.
4. Write program to demonstrate use of compile time and runtime polymorphism.
5. Write program to demonstrate use of friend function and friend class.
6. Write program to demonstrate use of virtual class.
7. Write program to demonstrate use of static data member and static member function.
8. Write program to demonstrate file handling.
9. Write program to demonstrate use of function templates.
10. Write program to demonstrate use of class templates.
11. Write program to demonstrate use of exception handling.
12. Write program to demonstrate command line arguments.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS203C.1	Develop logic of a program for solving real time problems and isolate and fix common errors in C++ programs.	6
BCS203C.2	Understand the object-oriented approach for the program development and make use of the OOP concepts (data abstraction, encapsulation, polymorphism, overloading, and inheritance) of C++ appropriately in problem solving.	2

BCS-107: Network Administration and Maintenance

Clock Hours: **30**

Course Objectives:

The objectives of the course are-

- To familiarize with evolution of networking and its type.
- To understand network hardware devices and topologies.
- To know about network setup, maintenance and fault finding.

Unit I: Evolution of Networking **L: 02**

- General
- Types of Networks
  - PAN (Personal Area Network)
  - LAN (Local Area Network)
  - MAN (Metropolitan Area Network)
  - WAN (Wide Area Network)
  - Local Area Network (LAN)
  - Wireless Local Area Network (WLAN)
  - Wide Area Network (WAN)
  - Virtual Private Network (VPN)
  - Virtual Local Network (VLAN)

Unit II: Network Hardware Devices L: 02

- Patch Panel
- Hub.
- Switch. L3, L2
- Router.
- Bridge.
- Gateway.
- Modem.
- Repeater.
- Access Point.
- Rack
- Firewall

Unit III: Networking Topologies L: 04

- Common network topologies:
  - Mesh,
  - Ring,
  - Bus,
  - Star
  - Tree

- Identifying Nodes in a Networked Communication
  - MAC Address
  - IP Address

Unit IV: Network Setup L: 06

- Networks Classes

Class	Default subnet mask	No. of networks	No. of host per network
A	255.0.0.0	256	16,777,214
B	255.255.0.0	65,536	65,534
C	255.255.255.0	16,77,216	126

- Internet, Web and the Internet of Things, The World Wide Web (WWW)
- Domain Name System (DNS)

Unit V: Network Maintenance L: 08

- Management of Network
  - Performance management and security management

- Maintenance Of Network
  - Preventative Maintenance
    - Physical care
    - Anti rat care
    - Cable maintenance
  - Hardware Maintenance
    - Physical fittings of switches
    - Rack mount
    - Up linking
  - Software Installation
    - Installation of Network controller Software like
    - Configuration of hardware's (Router, Firewall and switches)
    - Various software used for network

- Telnet
- Command Line Interface (CLI) through console port
- Web-based GUI
- Management of Network via manufacturers' Software's like (D-View 8, Enterprise Switch Manager)

Unit VI: Network Fault Finding **L: 08**

- Fault Finding of Network
  - Tracing of LAN node
  - Looping of networks
- Tools used for networking
  - Crimping Tool
  - Crone Tool
  - LAN Tester
  - Multi Meter
  - Soldering Iron
- Connectors and cables used in network
  - RJ 45
  - RJ 11
  - 9 PIN Connector
- Straight Cable
- Cross cable

References:

1. Behrouz A Forouzan, Data communications and Networking, Fourth Edition, Mc-Graw Hill Achyut Godbole, Data communications and Networks, TMH
2. W. Stallings, Data and Computer Communications‖, Eight Edition, Pearson Education.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS204C.1	Understand network administration and Maintenance	2

BCS-108: Word Processing Tools (Hands-on)

ClockHours: **30**

Course Objective:

The objectives of the course are-

- This course introduces word processing concepts and applications. Topics include preparation or a variety of documents and mastery of specialized software functions. Upon completion, students should be able to work effectively in a computerized word processing environment.

Unit I: Introduction to Word Processing **L: 06**

Development of the Word Processor, Design considerations for word processed documents, OpenOffice suite's word processing application/Writer, Opening and Closing Writer, Creating, opening and closing

documents, Working with multiple documents, Saving documents, Save an existing file under another name, Save different Versions.

Unit II: Locating the Components of the Writer Window **L: 06**

Writer toolbars and ruler, Adjusting the application settings, Adding content to Writer Documents, Working with text, Editing Content

Unit III: Formatting Documents **L: 05**

Text Formatting, Paragraph Formatting, Text alignment, Tabs and its types, Placing text at the tab position, Paragraph spacing, Working with lists, Paragraph borders and shading, Creating and Applying Styles

Unit IV: Concept of the Table **L: 05**

Adding data to a table, Deleting a table, Add and delete columns and rows, Modifying columns and rows, Images, Inserting images, Modifying images, Resize an image and charts

Unit V: Mail Merge **L: 04**

Preparing the documents, creating the main document, Creating the data source, Document formatting

Unit V: Google Docs **L: 10**

Create new documents from scratch, as well as from templates. Open existing documents from Google Docs, as well as other word processing programs. Navigate both the desktop and mobile versions of Google Docs. Format text, paragraphs, and pages. Insert, format, and edit images and graphics. Share and collaborate on documents with other users. Publish documents to the web. Print documents. Install add-ons to give Google Docs even more function and features.

Textbooks:

1. Nita Rutkosky and Audrey Rutkosky Roggenkamp, Microsoft Word 2013 with CD, Paradigm Publishing, 2011, ISBN 978-0-76385-199-6 (text and CD).
2. Bittu Kumar, Mastering MS Office, V&S Publishers, 2017

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS201C.1	The students will be able to explain the general concepts of the Word processors.	5
BCS201C.2	Apply navigation and format of business documents and envelopes, graphics, tables, charts, labels, themes, styles, forms, macros, and merging and sorting.	3
BCS201C.3	Create and use advanced features of macros, styles, graphics, document protection, table of contents, and indexes.	6

MTM-101:Calculus

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- This course aims to provide data science students with a solid understanding of calculus and its applications in data analysis, machine learning, and statistical modeling.
- The course will cover the key concepts of differential and integral calculus.
- Students will learn how to apply these concepts to solve real-world problems in data science.

Unit-I: Introduction to Calculus and Data Science **L: 02**

What is calculus and why is it important in data science?
Overview of data science applications that rely on calculus.

Unit-II: Limits and Continuity **L: 04**

Definition of limits and evaluating limits.
Continuity and its significance in data analysis.

Unit-III: Introduction to Calculus and Mathematical Foundations **L: 04**

Real numbers and the number line
Functions and their properties
Limits and continuity
Derivatives and their interpretations

Unit-IV: Differentiation **L: 06**

Techniques of differentiation (product rule, chain rule, quotient rule)
Applications of derivatives (optimization, rates of change, approximation)
Higher-order derivatives
Implicit differentiation

Unit-V: Integration **L: 05**

Antiderivatives and the indefinite integral
Techniques of integration (substitution, integration by parts)
Definite integrals and their properties
Applications of integration (area under curves, volumes, work)

Unit-VI: Sequences and Series **L: 05**

Convergence and divergence of sequences
Infinite series and their convergence
Power series and Taylor series

Unit-VII: Calculus for Data Science Applications **L: 04**

Linear regression and least squares
Optimization in machine learning
Calculus in probability and statistics
Calculus in data visualization

References:

1. Sangit Chatterjee and Bodhisattva Chattopadhyay, Calculus for Data Science: A Guide to Multivariable Analysis".
2. P. Kandasamy, K. Thilagavathy (2004), Mathematic for B.Sc. Vol.-I, II, III & IV, S. Chand& Company Ltd., New Delhi-55.
3. Shanti Narayan (2001) Differential Calculus. Shyamlal Charitable Trust, New Delhi.
4. Shanti Narayan (2001) Integral Calculus. S. Chand& Co. New Delhi.
5. S.Sudha (1998) Calculus. Emerald Publishers, Chennai.
6. G. B. Thomas and R. L. Finney. (1998) Calculus and Analytic Geometry, Addison Wesley (9thEdn.), Mass. (Indian Print) 6. P. R. Vittal. (2004) Calculus, Margham Publication, Chennai.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM101C.1	Understanding of calculus and its applications in data analysis, machine learning, and statistical modeling.	2
MTM101C.2	Learn how to apply the concepts of differential and integral calculus, to solve real-world problems in data science.	1

ETM-101: Digital Electronics

Clock Hours: **30**

Course Description:

- The Digital Electronics course is designed to provide students with a comprehensive understanding of the fundamental concepts and principles of digital circuits. The course covers various topics such as number systems, logic gates, Boolean algebra, combinational and sequential circuits, and digital logic design. Through a combination of theoretical instruction and practical exercises, students will develop the skills necessary to analyze, design, and troubleshoot digital circuits.

Course Objectives:

The objectives of the course are-

- To introduce students to the basic concepts and principles of digital electronics.
- To familiarize students with number systems and binary arithmetic.
- To provide an understanding of logic gates and Boolean algebra.
- To enable students to design and analyze combinational circuits.
- To develop students' skills in designing and working with sequential circuits.
- To develop problem-solving and critical thinking skills in the context of digital electronics.

Unit – I: Introduction to Digital Electronics **L: 04**

Basic concepts of digital electronics - Advantages of digital systems over analog systems - Digital signals and binary representation - Digital circuits and their applications

Unit – II: Number Systems and Binary Arithmetic **L: 06**

Decimal, binary, octal, and hexadecimal number systems - Conversion between number systems - Binary arithmetic operations: addition, subtraction, multiplication, and division – 1's and 2's complement representation and arithmetic

Unit – III: Logic Gates and Boolean Algebra **L: 06**

Basic logic gates: AND, OR, NOT, NAND, NOR, XOR, and XNOR - Truth tables and logic gate behaviour - Boolean algebra: laws and theorems - Simplification of Boolean expressions using Karnaugh maps and Boolean algebra

Unit – IV: Combinational Logic Design **L: 06**

Introduction to combinational circuits - Designing combinational circuits using logic gates - Multiplexers and demultiplexers - Encoders and decoders - Adders, subtractors, and magnitude comparators.

Unit – V:Sequential Logic Design **L: 08**

Introduction to sequential circuits - Flip-flops: SR, JK, D, and T flip-flops - Sequential circuit analysis and design - Registers and Counters

---

Reference Book

1. M. Morris Mano and Michael D Ciletti, Digital Design, Pearson Education.
2. Roger L. Tokheim, Digital Electronics: Principles and Applications.
3. Thomas L. Floyd, Digital Fundamentals.
4. Ronald J. Tocci, Neal S. Widmer, and Greg Moss, Digital Systems: Principles and Applications.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
ETM101C.1	Understand and apply number systems such as binary, decimal, octal and hexadecimal.	2
ETM101C.2	Analyze and design logic gates and comprehend Boolean algebra.	4
ETM101C.3	Design and implement combinational circuits using various building blocks.	6
ETM101C.4	Design and analyze sequential circuits including flip-flops, registers, and counters.	4
ETM101C.5	Demonstrate proficiency in troubleshooting digital circuits.	2
ETM101C.6	Apply problem-solving and critical thinking skills to real-world scenarios in digital electronics.	3

Practicals:

The practical component of the course is designed to provide hands-on experience and reinforce the theoretical concepts covered in the lectures. The practical sessions will allow students to work with actual electronic components, digital logic simulation software, and hardware description languages. The practicals will cover the following areas:

1. Number System Conversion: Practice converting numbers between binary, decimal, and hexadecimal representations.
2. Logic Gates: Construct and test logic gates using basic electronic components.

3. Boolean Algebra: Solve Boolean expressions and implement them using logic gates.
4. Combinational Circuit Design: Design and implement combinational circuits such as adders: Half and Full Adder, multiplexers, and decoders (Using Nand Gate: IC 7400 & 7410).
5. Sequential Circuit Design: Design and implement sequential circuits using flip-flops and state machines using IC 7476.

Through these practical exercises, students will gain practical skills and reinforce their understanding of digital electronics concepts.

DSM-101:Introduction to Data Science

*ClockHours:***30**

Course Description:

- Introduction to Data Science is a foundational course that provides students with a comprehensive understanding of the principles and techniques used in data science. The course aims to equip students with the necessary knowledge and skills to explore, analyze, and interpret data to make informed decisions and gain valuable insights. Throughout the course, students will be introduced to various data science concepts, tools, and methodologies to solve real-world problems.

Course Objectives:

The objectives of the course are-

- Define data science and explain its significance in various fields.
- Utilize data manipulation techniques to preprocess and clean data.
- Apply exploratory data analysis (EDA) methods to gain insights into datasets.
- Understand fundamental statistical concepts used in data science.
- Implement basic data visualization techniques to effectively communicate findings.
- Apply machine learning algorithms to solve prediction and classification tasks.
- Explain the importance of data ethics and privacy in data science projects.
- Work collaboratively on data science projects and communicate results effectively.

Unit 1: Introduction to Data Science **L: 02**

Definition and scope of data science
Data science applications in different domains
Introduction to data-driven decision making.

Unit 2: Data Collection and Preprocessing **L: 04**

Data types and sources
Data acquisition and storage
Data cleaning and handling missing values.

Unit 3: Exploratory Data Analysis (EDA) **L: 04**

Data summarization and descriptive statistics
Data visualization techniques
Identifying patterns and outliers in data.

Unit 4: Introduction to Statistics for Data Science **L: 04**

Probability distributions

Hypothesis testing
Correlation and regression analysis.

Unit 5: Data Visualization **L: 06**

Introduction to data visualization tools (e.g., Matplotlib, Seaborn)
Creating effective data visualizations
Design principles for data visualizations.

Unit 6: Introduction to Machine Learning **L: 06**

Supervised vs. unsupervised learning
Model training and evaluation
Common machine learning algorithms (e.g., linear regression, decision trees).

Unit 7: Ethics and Privacy in Data Science **L: 02**

Ethical considerations in data collection and analysis
Protecting sensitive information and ensuring privacy
Implications of biased data and algorithms.

Unit 8: Data Science Project **L: 02**

Forming teams and project planning
Data exploration and analysis
Presenting findings and conclusions.

References:

1. John D. Kelleher and Brendan Tierney, Data Science., ISBN 9780262535434,The MIT Press.
2. Wes Mc Kinney, Python for Data Analysis, O'Reilly.
3. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Introduction to Statistical Learning.
4. Jojo Moolayil, Smarter Decisions: The Intersection of IoT and Data Science, PACKT, 2016.
5. Cathy O'Neil and Rachel Schutt, Doing Data Science, O'Reilly, 2015.
6. David Dietrich, Barry Heller, Beibei Yang, Data Science and Big data Analytics, EMC 2013
7. Raj, Pethuru, Handbook of Research on Cloud Infrastructures for Big Data Analytics, IGI Global.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
DSM101C.1	Identify and describe the methods and techniques commonly used in data science.	3
DSM101C.2	Demonstrate proficiency with the methods and techniques for obtaining, organizing, exploring, and analyzing data.	2
DSM101C.3	Recognize how data analysis, inferential statistics, modeling, machine learning, and statistical computing can be utilized in an integrated capacity.	1

Detail Syllabus
(Semester III and IV)

Semester-III

BCS-203: Operating System

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Understand the basic components of Operating Systems and their interactions.
- Select the policies for Process Management, Memory Management and Deadlock Management.
- Understand the basics of File, Device and Disk Storage Management.

Unit I: **L: 06**

Operating System: Concept, Components of Operating System, Operating System Operations, Protection and Security. Computing Environment. **Abstract View of OS**: User view, System View, Operating System Services, **System Calls**: Concept, Types of System Calls. **Computer System Architecture**: Single-Processor Systems, Multiprocessor Systems. Types of Operating Systems: Batch Operating System, Multi-Programmed Operating System, Time-Shared Operating System, Real Time Operating System, Distributed Operating Systems.

Linux Operating System: Introduction to Linux OS, Basic Commands of Linux OS.

Unit II: **L: 08**

Process Concept, Operation on Processes, Cooperating Processes, Inter-Process Communication, Threads. **Process Synchronization**: Introduction, The Critical-Section Problem with solution, Bakery Algorithm, Synchronization hardware, Semaphores, Semaphores Implementation, Classical Problems of Synchronization with algorithms, Critical Regions, Monitors. **CPU Scheduling**: Basic Concepts, Scheduling Criteria, Scheduling algorithms, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling.

Linux Operating System: Process Management Commands and System Calls.

.**Unit III:** **L: 06**

Memory Management: Main Memory: Contiguous Memory Allocation, Fragmentation, Paging, And Segmentation. Virtual Memory: Demand Paging, Page Replacement, Page replacement algorithm, Allocation of frames, Thrashing.

Linux Operating System: Memory Management Commands and System Calls.

Unit IV: **L: 06**

File and Device management: File Concept, file protection, allocation methods, configuration, linked and index allocation. Directory structure – single level, two level, tree structure, cyclic graph and general graph directory, Free space management, device management, Disk structure, **disk scheduling**: FCFS scheduling, SSTF scheduling, SCAN scheduling, C-SCAN scheduling, Look scheduling, C-LOOK scheduling, disk scheduling algorithms.

Linux Operating System: File Management Commands and System Calls.

Unit V: Dead Locks **L: 04**

Modelling, characterization, prevention and avoidance, detection and recovery.

References:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne (2009), Operating System Concepts, 8th Ed., John Wiley ISBN 0-471-69466-5.
2. William Stallings (2014), Operating Systems: Internals and Design Principles. Pearson, 8th Ed. ISBN-13: 978-0-13-230998-1
3. Andrew S. Tanenbaum (2009), Modern Operating Systems, 3rd Ed., Pearson. ISBN: 0135013011
4. Andrew S. Tanenbaum, AS Woodhull (2006), Operating Systems Design and Implementation, 3rd Ed., Prentice Hall ISBN-10: 0131429388
5. M. J. Bach (1986), Design of the Unix Operating System, Prentice Hall of India ISBN0. -13-201757-1 025.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS211C.1	Understand the structure and functions of operating systems along with their components, types and working.	2
BCS211C.2	Make use of appropriate Linux commands for memory management, file management and directory management.	3
BCS211C.3	Analyze the performance of different scheduling algorithms along with the policies for concurrency and deadlock management.	4
BCS211C.4	Elaborate the system calls for process management and file management.	6

BCS-204: Data Structures

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To teach efficient storage mechanisms of data for an easy access.
- To design and implementation of various basic and advanced data structures.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structures.
- To improve the logical ability.

Unit I: **L: 10**

Data Structures – Introduction to Data Structures, abstract data types.

Stack ADT- definition, operations, array and linked implementations, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation.

Queue ADT- definition and operations, array and linked Implementations **Circular queues**-Insertion and deletion operations, Deque (Double ended queue) ADT, array and linked implementations.

Unit II: **L: 10**

Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, **circularly linked lists**- Operations for Circularly linked lists, doubly linked list implementation, insertion, deletion and searching operations, applications of linked lists.

Unit III: **L: 05**

Searching and Sorting – Searching-linear and binary search methods.

Sorting- selection sort, bubble sort, insertion sort, quick sort, merge sort comparison of sorting and searching methods.

Unit IV: **L: 05**

Trees – Definitions, tree representation, properties of trees, Binary tree, Binary tree representation, binary tree properties, binary tree traversals, binary tree implementation, applications of trees.

TEXT BOOKS:

1. E. Horowitz, S. Sahni and Susan Anderson Freed, Fundamentals of Data structures in C Universities Press, 2nd Edition.
2. D.S. Kushwaha and A. K. Misra, Data structures A Programming Approach with C, PHI., Kindle Edition.

REFERENCE BOOKS:

1. R. F. Gilberg and B. A. Forouzan, Data structures: A Pseudo code Approach with C, 2nd edition, Cengage Learning.
2. M. A. Weiss, Data structures and Algorithm Analysis in C, 2nd edition, Pearson.
3. Aaron.M.Tanenbaum,Y. Langsam, M.J.Augenstein, Data Structures using C,Pearson, ISBN: 9789332543546, 9332543542.
4. R. Kruse, C. L. Tondo and B. Leung, Data structures and Program Design in C, 2nd edition, Pearson.
5. Jean Paul Tremblay, Paul G. Sorenson, An introduction to data Structures with Applications, second edition, McGraw Hill Education.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS212C.1	Develop solutions for a range of problems using procedure oriented / object-oriented programming.	6
BCS212C.2	Choose the appropriate data structure and algorithm design method for a specified application.	3
BCS212C.3	Apply practical knowledge on the applications of data structure.	3

BCS-205: Lab on Data Structures

Course Objectives:

The objectives of the course are-

- Solve real-world problems by reasoning about data structure choices, choose appropriate implementations.
- To make the students write various programs and ADTS for all data structures.
- Students will learn to write, debug, and test large programs systematically.

Implementation of programs based on the following.

- Arrays
- Multidimensional Arrays, Matrices
- Stacks, Polish Notation
- Queues
- Deques
- Linear Linked List, Circular Linked List, Doubly Linked List
- Polynomial Addition/Subtraction

Implementation of programs based on Trees.

- Binary Search Tree
- In-order, Pre-order and Post-order Traversals
- Heap Tree

Implementation of programs for Searching and Sorting techniques

- Linear and Binary Search (using array)
- Bubble sort
- Selection sort
- Insertion sort
- Radix sort
- Quick sort
- Merge sort
- Heap sort.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS213C.1	Develop solutions for a range of problems using procedure oriented / object-oriented programming.	2
BCS213C.2	Choose the appropriate data structure and algorithm design method for a	3

	specified application.	
BCS213C.3	Apply practical knowledge on the applications of data structure.	3

BCS-206:Lab on Linux administration

Course Objectives:

The objectives of the course are-

- Perform Linux Operating System Installation.
- Implement the policies of Process Management, Inter process communication and MemoryManagement.
- Apply the basics of File Management, Device and Disk Storage Management.

Demonstration on Installation of Linux system

Direct Installation: Partitioning the Hard drive for Linux, Using Live CD, Virtual Machine, init and run levels

Linux Commands and Shell Programming

Creating Users Accounts and Groups, Starting and Stopping Services, Files and File System (File Types and Permissions, Links, Size and Space, Date and Time), Working with Files: Reading Files, Searching for files, Copying, Moving, Renaming, Deleting, Linking, and Editing Files, Other Commands: ls, rm, rmdir, pwd, more, less. grep, sort, cat, head, tail, wc, tee, ps, top, tar, unzip, nice, kill, netstat, Disk related commands, checking disk free spaces read command, conditional and looping statements, case statements, parameter passing and arguments, Shell variables, system shell variables, shell keywords, Shell programs for performing various tasks (List to be given by the course instructor)

System Administration

Managing user accounts-adding & deleting users, changing permissions and ownerships, Creating and managing groups, modifying group attributes, Temporary disable user's accounts, creating and mounting file system, checking and monitoring system performance file security &Permissions, becoming super user using su; Getting system information Backup and restore files, reconfiguration hardware with kudzu, installing and removing packages in Linux. X-Windows administration

---

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS214C.1	Implement the Installation of Linux system.	6
BCS214C.2	Understand the basic commands of Linux operating system and can write shell scripts.	2
BCS214C.3	Implement system administration tasks.	6

MTM-102: Graph Theory

ClockHours: **60**

Course Objectives:

The objectives of the course are-

- To introduce the fundamental theory of graphs and their homomorphisms.
- To understand directed graphs and graph coloring in detail.
- To know about special classes of graphs with planarity.

Unit I: **L: 15**

Introduction to graph theory; Subgraphs, applications of graph; finite and infinite graphs; types of graph, incidence and degree; isolated vertex; pendant vertex; null graph; paths, cycles, and trails; Eulerian circuits, Other important terminology, Graph isomorphism, operations on graph

Unit II: **L: 20**

Representation of graphs, Special classes of graphs: Bipartite graphs, line graphs, chordal graphs, connected graphs and shortest paths: Walks, trails, paths, connected graphs, distance, cut-vertices, cut-edges, blocks, connectivity, weighted graphs, shortest path algorithms, traversing algorithm of graphs

Unit III: **L: 15**

Directed Graphs -Types of Directed Graphs, Digraphs and Binary Relations, Directed Paths and Connectedness, Euler Graphs, Planarity: Planar graphs, Euler's formula, Graphs on other surfaces ,Dual graphs, Infinite graphs

UNIT IV: **L: 10**

Graph Coloring: Colouring graphs, Definition of Chromatic number. Chromatic Partitioning. Chromatic Polynomial, Colouring vertices, Brooks' theorem, Colouring maps, Colouring edges

---

REFERENCE BOOKS:

1. J.A. Bondy and U.S.R. Murty: Graph Theory and Applications (Freely downloadable from Bondy's website; Google-Bondy).
2. D.B. West: Introduction to Graph Theory, Prentice-Hall of India/Pearson, 2009 (latest impression).
3. Narsingh Deo, Graph Theory with Applications to Engineering & Computer Science, Dover Publications, INC. Mineola, New York.
4. R. Diestel: Graph Theory, Springer(low price edition) 2000.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM212C.1	Understand terminology of graphs, graph isomorphism.	2
MTM212C.2	Apply the fundamental concepts of Graphs and their applications.	3
MTM212C.3	Analyze the subgroups of special classes of graphs, connected graphs	4

	and shortest path.	
MTM212C.4	Explain the significance of the graph coloring and planarity.	2

ETM-102: Microprocessor Basics and Programming

*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- This course introduces internal architecture, programming model of Intel Microprocessors (8086-Pentium) and assembly language programming using an assembler. Students will also learn the interfacing of memory and I/O devices with microprocessor.

Unit I: **L: 10**

Microprocessor architecture: Internal architecture, Programming Model, Addressing modes, Data movement instructions.

Unit II: **L: 10**

Microprocessor programming: Register Organization, instruction formats, Program control instructions, assembly language.

Unit III: **L: 10**

Interfacing: Bus timings, Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, timer, Interrupt controller, DMA controller, video controllers, communication interfaces.

Unit IV: **L: 10**

Data transfer schemes: Synchronous data transfer, asynchronous data transfer, interrupt driven data transfer, DMA mode data transfer.

Unit IV: **L: 10**

Microprocessor controllers: I/O controllers, interrupt controller, DMA controller, USART controller.

Unit V: **L: 10**

Advance microprocessor architecture: CISC architecture, RISC architecture, superscalar architecture, multicore architecture.

REFERENCE BOOKS:

1. Brey, B.B.(2009). The Intel Microprocessors: Architecture, Programming and Interfacing. 8thedition. Pearson Education.
2. Triebel, W.A., & Singh, A. (2002). The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware and Applications. 4th edition. Pearson Education.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive

		level
ETM212C.1	Describe the internal architecture of Intel microprocessors.	2
ETM212C.2	Define and implement interfaces between the microprocessor and the devices.	6
ETM212C.3	Write assembly language programs.	6

BCS-207:Web Design-I

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To understand the basics of web development.
- To learn the concepts of HTML 5.0.
- To understand the importance of style sheet and able to apply the internal and external style sheet to format the web pages.
- To learn the basic concepts of JavaScript and Bootstrap.

Unit I: **L: 8**

Introduction to Web Development: Components of Web applications, Static vs. dynamic web pages, A survey of browsers, servers and scripting languages, URL, Introduction to HTML, HTML5, XHTML and CSS. HTML 5.0: Structure of HTML5 Document, Elements, tags and attributes, DOCTYPE declaration, HEAD section, Coding text elements, Headings and paragraphs, Inline elements for formatting and emphasizing text, div and span elements, Events.

Unit II:

L: 12

CSS to format elements of a web page: CSS applications, Selectors: Basic Rule, Grouping, class and ID selectors, attribute selectors, document structure, Specificity, Inheritance, cascade, Values and Units, Fonts, Colors Working with text, text alignment, spacing(letter/word), Text decoration, Text Shadows, CSS box model, Spacing, borders and backgrounds, Page layout using CSS, Floating elements, Multicolumn layout, Ways to include CSS in a web page.
HTML and CSS skills: Working with links and lists, working with images, working with tables, working with forms, Adding audio, video to web pages.

Unit III: **L:10**

JavaScript: Variable declaration, Operators, Control Statements, Error Handling, understanding arrays, Function Declaration, Built in Functions, Standard Date and Time Functions, JavaScript for image rollovers, image swaps, slide shows, tabbed data and data validation.JS Objects, JS Array, JS String, JS OOPs , JS Events, JS Exception handling, JS Typed Array, JS Set, JS Map, Real World Applications of JavaScript: Creating sliding menus, Creating pop-up menus. Bootstrap: Overview, Component and utilities.

REFERENCE BOOKS:

1. Anne Boehm, Murach's HTML, XHTML and CSS, Shroff Publishers and Distributers Pvt. Ltd., ISBN 13: 978-93-5023-095-4 2

2. Eric A. Meyer, CSS: The Definitive Guide, Visual Presentation for the Web, 3rd Edition, O'Reilly Media, November 2006, ISBN:978-0-596-52733-4| ISBN 10:0-596-52733-0(Print)
3. Jake Spurlock, Bootstrap, O'Reilly publication.
4. Bootstrap Tutorial, Learning by tutorialspoint.com.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS215C.1	Design and implement web pages.	6
BCS215C.2	Design web forms and apply client-side validation.	6
BCS215C.3	Demonstrate various CSS features.	2
BCS215C.4	Create a drawing application with Canvas using HTML5.	6
BCS215C.5	Display the location's coordinates of Longitude and latitude on Google Map.	6
BCS215C.6	Utilize the bootstrap JavaScript Plugins to develop modern web pages.	3

Semester-IV

BCS-208: Database Management System(DBMS)

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Introduction to the basic concepts of database management systems, learning to design databases using ER modelling, and decomposing data based on functional dependencies.
- Understand Relational databases, SQL, Transaction management, Query processing, concurrency control and recovery system.

Unit I: Database system architecture **L: 05**

Data Abstraction, Data Independence, Data Definition and Data Manipulation Languages.

Unit II: Data models **L: 05**

Entity-relationship, network, relational, hierarchical and object-oriented data models, integrity constraints and data manipulation operations.

Unit III: Relational Databases **L: 10**

Relational query languages Relational algebra, tuple and domain relational calculus, SQL and QBE.
Relational database design Codd's Rules, Domain and data dependency, Armstrong's axioms, normal forms, dependency preservation, lossless design.

Unit IV: Query processing and optimization **L: 05**

Evaluation of relational algebra expressions, query equivalence, join strategies, query optimization algorithms.

Unit V: Transaction processing **L: 05**

Recovery and concurrency control, locking and timestamp-based schedulers, multi-version and optimistic Concurrency Control schemes.

REFERENCE BOOKS:

1. Michael Kifer, Arthur Bernstein, P.M, Lewis and P.K. Panigrahi (2011), "Database Systems: An Application Oriented Approach", Second Edition, Pearson Education, 2011, ISBN:9788131703748.
2. C. J .Date, A. Kannan and S. Swamynathan (2006), "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006, ISBN:978-81-7758-556-8.
3. Silberschatz, H.F. Korth, and S. Sudarshan (2011), "Database System Concepts", TMH Publications, Sixth Edition, 2011, ISBN: 978-007-132522-6.
4. Ramez Elmasri, Shamkant B. Navathe (2011), "Fundamentals of Database Systems" Seventh Edition, Pearson Education, 2011, ISBN: 978-0-13-397077-7.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS221C.1	Apply the relational model, specify integrity constraints.	3
BCS221C.2	explain how to create a relational database using an ER diagram and normalization techniques	2
BCS221C.3	Apply SQL to create, query and manipulate relational databases.	3
BCS221C.4	Determine partitioning and distribution of data across networked nodes of a DBMS and data optimization in a distributed environment.	5

BCS-209: Java Programming

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Understand Fundamental concepts of object-oriented programming using Java technology.
- Java applications development using polymorphism, inheritance, interfaces and inner classes and multi-threading.
- GUI applications and event driven applications development.

Unit I: An Introduction to Java **L: 05**

History of Java, Features of Java (Java Buzz words), Obtaining Java Environment, setting up Java Environment, Structure of the Java Program, creating a Source File, Compiling the Source File into a .class file, Executing the Program, The Java Virtual Machine, Comments, Data types, variables, Keywords, Operators, type casting, Control Structures, Arrays, command line argument.

Unit II: Introduction to OOPs **L: 05**

OOPs concepts, Predefined classes(String, StringBuffer), type casting, wrapper classes, Input and Output, User defined class, object creation and initialization, finalize ()method, static fields and methods, this keyword, Access specifier, Inner class.

Unit III: Inheritance, Polymorphism and interfaces **L: 05**

Dynamic Polymorphism (Method Overloading and Method Overriding), Static Polymorphism, final keyword, Superclass, Subclass, super keyword, Abstract classes, Methods with a Variable Number of Parameters, Interfaces

Unit IV: Multithreading and Exceptions **L: 05**

Creating Thread, Multi-Tasking using Threads, Thread Synchronization or Thread Safe, Thread Class Methods, Thread Communication, Thread Properties, Thread Group, Thread States (Life-Cycle of a Thread), Exception handling (try, catch, finally), Types of Exceptions(built-in, user defined)

Unit V:Graphics Programming and event handling **L: 06**

Introduction to swing and awt, Creating a Frame, Positioning a Frame, Displaying Information in a Component, Working with 2D Shapes, Color, Special Fonts for Text, JComponent class Methods, Creating Components in Swing (PushButton, Label, JComboBox Class , JList Class, JMenu Class),

Layout Manager (FlowLayout, BorderLayout, CardLayout, GridLayout, GridBagLayout), Basics of Event Handling, Listeners andListener Methods, Mouse Events, Keyboard Events, Window Event, AWT Event Hierarchy.

Unit VI: Streams, Files and JDBC **L: 04**

Input and output stream, Reading and Writing text Data, File Management(File Class), The Design of JDBC, JDBC Configuration, Executing SQL Statements.

REFERENCE BOOKS:

1. Horstman Cay, Cornell Gary, Core JavaTM2, Vol.1&2, Seventh Edition, Pearson education.
2. Herbert Schildt, The Complete Reference, Seventh Edition, Tata McGraw-Hill.
3. Steven Holzner, JAVA 2 Programming Black Book, Wiley India.
4. Ivor Horton, Beginning Java 2, JDK 5 Ed, Wiley India.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS222C.1	Create Java application development using polymorphism, inheritance, and inner classes.	6
BCS222C.2	Develop GUI interface and event driven applications.	6
BCS222C.3	Manipulate databases through java application.	6

BCS-210:LAB on Java Programming

Course Objectives:

The objectives of the course are-

- Programming using inner classes and inheritance, polymorphism and interfaces.
- Use various swing components and handle several events in the development of GUI applications.
- Use JDBC and package creation.

---

1. Write a program that demonstrates program structure of java with use of arithmetical and logical implementation.
2. Write a program that demonstrates string operations using String and StringBuffer class.
3. Write a program to demonstrate inner class and static fields.
4. Write a program that demonstrates inheritance, polymorphism.
5. Write a program that demonstrates 2D shapes on frames.
6. Write a program that demonstrates color and fonts.
7. Write a program to illustrate the use of various swing components.
8. Write a program that demonstrates use of dialog box and menus.
9. Write a program that demonstrates event handling for various types of events.
10. Write a program to illustrate multithreading.
11. Write a program to illustrate exception handling.
12. Write a program to demonstrate the use of File class.

13. Write a program that demonstrates JDBC on application.
14. Write a program that demonstrate package creation and use in program.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS223C.1	Understand Object oriented programming concepts using Java.	2
BCS223C.2	Knowledge of input, its processing and getting suitable output.	2
BCS223C.3	Understand, design, implement and evaluate classes and applets.	6
BCS223C.4	Knowledge and implementation of AWT package	6

BCS-211:LAB on DBMS (MySQL)

Course Objectives:

The objectives of the course are-

- Provides foundation knowledge in database concepts, technology, and practice to prepare students into expert database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

---

1. Creating database tables and using data types.
   Create table, Modify table, Drop table.
2. Practical Based on Data Manipulation.
   Adding/Modify/Delete data using Insert/ Update/ Delete
3. Practical Based on Implementing the Constraints.
   NULL and NOT NULL, Primary Key Constraint, Foreign Key Constraint
   Unique Constraint, Check Constraint, Default Constraint
4. Practical for Retrieving Data Using following clauses.
   Simple select clause.
   Accessing specific data with Where Clause
   Ordered By/ Distinct/Group By Clause.
5. Practical Based on Aggregate Functions.
   AVG, COUNT, MAX, MIN, SUM, CUBE
6. Practical Based on implementing all String functions.
7. Practical Based on implementing Date and Time Functions.
8. Practical Based on implementing use of UNION, INTERSECTION, SET DIFFERENCE.
9. Implement Nested Queries & all types of JOIN operation.
10. Practical Based on performing different operations on a view.
11. Practical Based on implementing use of Procedures.
12. Practical Based on implementing use of Triggers.
13. Practical Based on implementing Cursor.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS224C.1	Design and implement a database schema for a given problem-domain.	6
BCS224C.2	Create and maintain tables using PL/SQL, Populate and query a database using SQL DML/DDL commands and programming PL/SQL including stored procedures, stored functions, cursors, triggers.	6

MTM-103: Mathematics for Machine Learning

ClockHours: **60**

Course Objectives:

The objectives of the course are-

- To appreciate ML beyond a black box and also to analyze, design and improve them.
- Finally this specialization aims to give the students the prerequisite mathematical knowledge to continue their journey and take more advanced courses in machine learning.

Unit I: Linear Algebra **L: 05**

Matrices, Vector Space, Basis and Rank, Linear Mappings, Affine Spaces.

Unit II: Analytic Geometry **L: 05**

Norms, Inner Products, Distances, Orthogonality, Orthonormal Basis, Complement, Inner Products, Orthogonal Projections, Rotations.

Unit III: Matrix Decompositions **L: 05**

Eigenvalues, Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, SVD, Matrix Approximation and Phylogeny

Unit IV: Vector Calculus **L: 05**

Partial Differentiation and Gradients, Gradients of Vector-valued functions, and Matrices. Backpropagation and automatic differentiation, Higher order derivatives, Linearization and Multivariate Taylor Series.

Unit V:Probability and Distributions and Sampling Theory **L: 10**

Probability Space, Baye's Theorem, Summary Statistics and Independence, Gaussian Distribution, Conjugacy and the Exponential family, Change of variables and/ inverse transform, Central Limit Theorem, Inequalities.

Unit VI: Continuous Optimization **L: 05**

Optimization using Gradient Descent, Constrained Optimization and Lagrange Multipliers, Convex Optimization.

Unit VI: Applications to Machine Learning Problems **L: 25**

Data and Models: Data, Models and Learning. Empirical Risk Minimization, Parameter Estimation, Probabilistic Modeling and Inference, Directed Graphical Models, Model Selection, Linear Regression and Gaussian Process.

Dimensionality Reduction with PCA: Maximum variance and Projection perspective, Eigenvector computation and low-rank approximation, PCA in higher dimensions. Linear Discriminant Analysis.

Density Estimation with Gaussian Mixture Models: Parameter Learning via Maximum Likelihood, EM algorithm, Latent-variable Perspective.

Classification with Support Vector Machines: Separating hyperplanes, Primal SVM, Dual SVM, Kernels, Numerical solutions.

REFERENCE BOOKS:

Textbook:

1. Deisenroth, Marc Peter, A. Aldo Faisal, and Cheng Soon Ong. Mathematics for machine learning. Cambridge University Press, 2020.

Reference Books:

1. Trivedi, Kishor Shridharbhai. Probability and statistics with reliability, queuing, and computer science applications. Vol. 2. New York: Wiley, 2002.
2. Strang, Gilbert, Gilbert Strang, Gilbert Strang, and Gilbert Strang. Introduction to linear algebra. Vol. 3. Wellesley, MA: Wellesley-Cambridge Press, 1993.
3. Lipschutz, Seymour, Murray R. Spiegel, and Dennis Spellman. Vector analysis and an introduction to tensor analysis. McGraw-Hill, 2009.
4. Levin, Richard I. Statistics for management. Pearson Education India, 2011.
5. Bishop, Christopher M. Pattern Recognition and Machine Learning by Christopher M. Bishop. Springer Science+ Business Media, LLC, 2006.
6. Theodoridis, Sergios, and Konstantinos Koutroumbas. "Pattern recognition. 2003." Google Scholar Google Scholar Digital Library Digital Library (2009).

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM223C.1	Appreciate the fundamentals of linear algebra, vector calculus, probability and optimization as applied to machine learning. (Understanding)	5
MTM223C.2	Analyze the ML algorithms for error and efficiency. (Applying, Analyzing, Evaluating)	4
MTM223C.3	Design and analyze solutions for real life problems employing ML algorithms. (Analyzing, Creating).	6

ETM-201: Microcontroller

Clock Hours: **60**

Course Objectives:

The objectives of the course are-

- Understand the difference between a Microprocessor and a Microcontroller and embedded microcontrollers.
- Familiarize the basic architecture of 8051 microcontroller.
- Program 8051microprocessor using Assembly Level Language and C.
- Understand the interrupt system of 8051 and the use of interrupts.

Unit I: Introduction of Microcontrollers **L: 10**

Introduction to microcontrollers, difference in controller and processor. Architecture of 8051, Internal block diagram, Internal RAM organization, SFRS, pin diagram of 8051, I/O ports structure& operation, External Memory Interface.

Unit II: Assembly language Programming **L: 15**

Instruction classification, Instruction set, Addressing Modes: Immediate, register, direct, indirect and relative, assembler directives - features with example, I/O Bit & Byte programming using assembly language for LED and seven segment display(SSD) interfacing.

Unit III: Timer / Counter, Serial communication, Interrupts **L: 20**

TMOD, TCON, SCON, SBUF, PCON Registers, Timer modes, programming for time delay using mode1 and mode2. Introduction to interrupt, Interrupt types and their vector addresses, Interruptenable register and interrupt priority register(IE,IP), Synchronous and asynchronous serial communication

Unit IV:I/O Interfacing **L: 15**

Introduction 8051 programming in C, Programming serial port without interrupt, Use of timer to select baud rate for serial communication. Interfacing ADC, DAC, LCD, stepper motor.

REFERENCE BOOKS:

Reference Books:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi and McKinley, 8051 microcontroller and Embedded system using Assembly and C, Pearson Education, 2nd Edition 2008.
2. K.Uma Rao and Andhe Pallavi, The 8051 microcontroller – Architecture, programming and applications:, Pearson publications, First Edition 2010.
3. Myke Predko, Programming and Customizing the 8051 Microcontroller, Tata McGraw-Hill Publishing Company Ltd, Tata McGraw-Hill Edition 1999.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
ETM223C.1	Understand Architecture of Intel 51 family Microcontroller.	2
ETM223C.2	Interface I/O devices to Microcontroller.	6
ETM223C.3	Acquire the skill of writing assembly / C language programs.	6
ETM223C.4	Understand Serial communication using Microcontroller	2

BCS-212: Data Engineering

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To understand the basics of database development.
- To learn the concepts of MongoDB.
- To understand the importance of SQL.
- To learn the difference between SQL and NoSQL database.
- To design, deploy, and manage a highly available Postgres database using EDB Postgres Distributed (PGD).

Unit I: **L: 15**

Introduction to NoSQL Database: What in NoSQL? Difference between NoSQL and RDBMS, Benefits of NoSQL, EDB PostgreSQL: Overview, environment, syntax, data types, PostgreSQL queries, operators, clauses, Advanced PostgreSQL: Constraints, Joins, Union clause, Triggers, Views, Indexes, Alter table command, Transactions, Locks, Sub queries, Privileges, Date and Time functions and operators, PostgreSQL interface.

Unit II: **L: 15**

Introduction & Overview of MongoDB: Objectives, Design Goals, The Mongo Shell, JSON Introduction, JSON Structure, MongoDB Installation: Installing Tools, Overview of Blog Project, Swig, Express, Node Packaged Modules (npm), CRUD Operation in MongoDB: CRUD (Creating, Reading & Updating Data) Mongo Shell, Query Operators, Update Operators and a Few Commands.

REFERENCE BOOKS:

- Manu Sharma, MangoDB: Complete Guide, , BPP Publication
- Korry Douglas, Susan P. Douglas, PostgreSQL, ISBN : 9780672327568, 0672327562,Sams Publication.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS212C.1	Implement various types of operations in MangoDB.	6
BCS212C.2	Understand MangoDB functions and its implementation.	2
BCS212C.3	To learn about enterprise class tools for Postgres administration and prepares them to handle real time, production environments.	2

Detail Syllabus

Data Science

(Semester V and VI)

Semester-V

BDS-301: Design and Analysis of Algorithms

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Basic concepts of algorithms and analyze the performance of algorithms.
- Algorithm design techniques for developing algorithms.
- Searching and traversal algorithms for graphs.
- Nondeterministic algorithms and NP class of problem.

Unit I: **L: 06**

Review of Data Structures

Program Performance: Time and space complexity, asymptotic notation, complexity analysis, recurrence equations and their solution.

Unit II: **L: 12**

Algorithmic Techniques: Algorithm design strategies, divide and conquer, merge sort, quick sort and its performance analysis, randomized quick sort, Strassen's matrix multiplication; Greedy method and its applications, knapsack problem; Dynamic programming and its performance analysis, optimal binary search trees, 0/ 1 knapsack problem; Traveling salesman problem; Back-tracking, n-queens problem, graph coloring, Hamiltonian cycles, Branch and bound examples, 15-puzzle problem.

Unit III: **L: 08**

Graph Algorithms: DFS and BFS, spanning trees, biconnectivity; Minimum cost spanning trees: Kruskal's, Prim's and Sollin's algorithms; Path finding and shortest path algorithms; Topological sorting; Bipartite graphs.

Unit IV: **L: 04**

Infeasibility: P and NP-classes, NP-hard problems, reduction.

REFERENCE BOOKS:

1. Horowitz E. and Sahni S. Fundamentals of computer Algorithms, Galgotia publications.ISBN:0716783169.
2. Horowitz E., Sahni S. and RajasekaranS (),Computer Algorithms, Computer Science Press, ISBN-10: 8173716129.
3. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani(2006), Algorithms. McGraw-Hill publications, ISBN 9780073523408
4. Cormen, Leiserson and Rivest, Introduction to Algorithms, Prentice Hall of India, ISBN:978-81-203-4007-7

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS311C.1	Analyze the asymptotic performance of algorithms.	3
BDS311C.2	Write rigorous correctness proofs for algorithms.	6
BDS311C.3	Design and analyze divide-and-conquer based algorithms.	6
BDS311C.4	Devise and synthesize greedy and dynamic-programming based algorithms.	3
BDS311C.5	Employ graphs to model problems solvable using traversal techniques.	3
BDS311C.6	Able to model problems using backtracking.	3
BDS311C.7	Able to classify nondeterministic polynomial time algorithms.	2

BDS-302: Software Development Methodology

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- To introduce git for software development
- To learn the principles and practices associated with each of the agile development methods.
- To apply the principles and practices of agile software development on a project of interest and relevance to the student.

Unit I: **L: 08**

Git & Version Control: Basic Concepts, Environment setup, Life Cycle, Branches & Merging, working with local repository and Remote Repository.

Unit II: **L: 12**

Agile Methodology: Introduction, software development with agile, traditional model vs agile model, agile methods classification, manifesto and principles, project management, team interactions, ethics in teams, agility in design and testing, documentations, agile drivers, capabilities and values.

Unit III: **L: 08**

Agile Processes: Lean production – SCRUM, Crystal, Feature Driven Development, Adaptive Software Development, and Extreme Programming: Method overview, lifecycle, work products, roles and practices.

REFERENCE BOOKS:

1. Robert C. Martin ,Agile Software Development, Principles, Patterns, and Practices Alan Apt Series.
2. Succeeding with Agile : Software Development Using Scrum, Pearson.

3. www.github.com

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS312C.1	Use git for software development and deployment.	3
BDS312C.2	Apply a thorough understanding of Agile principles and specific practices.	3
BDS312C.3	Judge, craft and evaluate appropriate adaptations to existing practices or processes depending upon analysis of typical problems	5

BDS 303: Artificial Intelligence

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- This course introduces the basic concepts and techniques of Artificial Intelligence (AI).
- The course aims to introduce intelligent agents and reasoning, heuristic search techniques, gameplaying, knowledge representation, reasoning with uncertain knowledge.

Unit I: **L: 05**

Introduction, Definition, Future of Artificial Intelligence, characteristics of Intelligent Agents, Typical Intelligent Agents, Problem Solving Approach to Typical AI problems.

Unit II: **L: 10**

Problem solving Methods, Search Strategies, Uninformed, Informed, Heuristics, Local Search Algorithms and Optimization Problems, Searching with Partial, Observations, Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search, Game Playing, Optimal Decisions in Games, Alpha, Beta Pruning, Stochastic Games.

Unit III: **L: 10**

Knowledge Representation First Order Predicate Logic, Prolog Programming, Unification, Forward Chaining, Backward Chaining, Resolution, Knowledge Representation, Ontological Engineering, Categories and Objects, Events, Mental, Events and Mental Objects, Reasoning Systems for Categories, Reasoning with default Information.

UNIT IV: **L: 05**

Software Agents Architecture for Intelligent Agents, Agent communication, Negotiation and Bargaining, Argumentation among Agents, Trust and Reputation in Multi-agent systems, Applications AI applications

---

REFERENCE BOOKS:

1. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, Third Edition, 2009.
2. Stuart Russell, Peter Norvig Artificial Intelligence: A Modern Approach, 4th Edition, University of California at Berkeley, Pearson education, 2020.
3. Bratko, Prolog: Programming for Artificial Intelligence, Fourth Edition, Addison-Wesley Educational Publishers Inc., 2011
4. M. Tim Jones, Artificial Intelligence: A Systems Approach (Computer Science),Jones and Bartlett Publishers, Inc.; First Edition, 2008.
5. Nils J. Nilsson, The Quest for Artificial Intelligence, Cambridge University Press, 2009.
6. Gerhard Weiss, Multi Agent Systems, Second Edition, MIT Press, 2013.
7. David L. Poole and Alan K. Mackworth, Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS313C.1	Identify problems that are amenable to solution by specific AI methods.	3
BDS313C.2	Represent knowledge in Prolog and write code for drawing inferences.	6
BDS313C.3	Identify appropriate AI technique for the problem at hand.	3
BDS313C.4	Compare strengths and weaknesses of different artificial Intelligence techniques.	4
BDS313C.5	Sensitive towards development of responsible Artificial Intelligence.	6

BDS-304:Lab on DAA

Course Objectives:

The objectives of the course are-

- To convert the algorithms to code.
- To measure the complexities at run time.
- To modify the algorithms for efficiency.
- To debug and test the programs.
- To conclude using profile of outcomes.

---

1. Implement Insertion Sort (The program should report the number of comparisons)
2. Implement Merge Sort(The program should report the number of comparisons)
3. Implement Heap Sort (The program should report the number of comparisons)
4. Implement Randomized Quick sort (The program should report the number of comparisons)
5. Implement Radix Sort
6. Create a Red-Black Tree and perform following operations on it:
   a. Insert a node
   b. Delete anode
   c. Search for a number & also report the color of the node containing this number.
7. Write a program to determine the LCS of two given sequences

8. Implement Breadth-First Search in a graph
9. Implement Depth-First Search in a graph
10. Write a program to determine the minimum spanning tree of a graph
11. For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS314C.1	Able to construct logic for algorithms designed using designing techniques.	3
BDS314C.2	Able to do posterior analysis of the algorithms.	4
BDS314C.3	Able to debug the algorithms.	4
BDS314C.4	Modify to improve performance of the algorithms.	6
BDS314C.5	Able to test and profile the algorithms.	4

BCS 212: Introduction to Big Data Analytics (elective)
(Swayam)
(noc23-cs112)
*ClockHours:***60**

BCS 213:Computer Graphics (elective)
(Swayam)
(noc23-cs115)
*ClockHours:***60**

MTM-104:Linear Algebra
*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- Course provides mathematical material in Linear Algebra foundational for mathematics, engineering, and the sciences.
- It focuses on linear equations, matrix methods, analytical geometry and linear transformations.

Unit I: Systems of Linear Equations **L: 05**

System of Linear Equations, Augmented Matrix, Gaussian Elimination with Back-Substitution, Gauss-Jordan Elimination, Solving a Homogeneous System of Linear Equations.

Unit II: Matrices **L: 05**

Operations with Matrices: Addition, Scalar Multiplication, Matrix Multiplication, Identity, Transposes, Row operations, Finding the Inverse of a Matrix by Row Operations.

Unit III: Determinants **L: 10**

Determinant of 2x2 and 3x3 Matrices, Minors and Cofactors of a Square Matrix, Definition of the Determinant of a Square Matrix (Cofactor Theorem), Properties of Determinants, Cramer's Rule.

Unit IV: Vector Spaces **L: 10**

Vector Space, Subspace, Linear Combination, Linear Independence, Basis, Dimension, finding a Basis of a Vector Space, Coordinates, Change of Basis.

Unit V:Inner Product Spaces **L: 10**

Inner Product, Length, Orthogonal Vectors, Triangle Inequality, Cauchy-Schwarz, Inequality, Orthonormal Orthogonal) Basis, Gram-Schmidt Process.

Unit VI: Linear Transformations **L: 10**

Linear Transformations and Matrices for Linear Transformation, Kernel and Range of a Linear Transformations, Change of Basis.

Unit VII: Eigenvalues and Eigenvectors **L: 10**

Eigenvalues and Eigenvectors, Definition of Eigenvalue and Eigenvector, Diagonalization, Symmetric Matrices and Orthogonal Diagonalization.

REFERENCE BOOKS:

1. Ron Larson, Calculus: Elementary Linear Algebra by, 8th edition, Cengage Learning,2017.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM314C.1	Employ techniques to classify and solve linear systems of equations.	3
MTM314C.2	Illustrate the use of matrices and determinants.	2
MTM314C.3	Utilize vector spaces and linear transformations.	3
MTM314C.4	Explore the concept of orthogonality in vector spaces.	5
MTM314C.5	Compute eigenvalues and eigenvectors of matrices	5

ETM-202:Internet of Things

*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- The objective of this course is to impart necessary and practical knowledge of components of the Internet of Things.
- Develop skills required to build real-life IoT based projects.

Unit I: Introduction to IoT **L: 15**

Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing, Actuation, Basics of Networking, M2M and IoT Technology Fundamentals- Devices and gateways, Data management, Business processes in IoT, Everything as a Service(XaaS), Role of Cloud in IoT, Security aspects in IoT.

Unit II: Elements of IoT **L: 15**

Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces. Software Components- Programming API's (using Python/Node.js/Arduino) for Communication Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP.

Unit III: IoT Application Development **L: 15**

Solution framework for IoT applications- Implementation of Device integration, Data acquisition and integration, Device data storage- Unstructured data storage on cloud/local server, Authentication, authorization of devices.

Unit IV: IoT Case Studies **L: 15**

IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation.

REFERENCE BOOKS:

1. Vijay Madisetti, Arshdeep Bahga, Ïnternet of Things, A Hands on Approach, University Press.
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, Introduction to Internet of Things: A practical Approach, ETI Labs.
3. Pethuru Raj and Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press.
4. Jeeva Jose, Internet of Things‖, Khanna Publishing House, Delhi.
5. Adrian McEwen, Designing the Internet of Things, Wiley.
6. Raj Kamal, Internet of Things: Architecture and Design, McGraw Hill
7. Cuno Pfister, Getting Started with the Internet of Things, O Reilly Media.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
ETM314C.1	Understand internet of Things and its hardware and software components	2
ETM314C.2	Interface I/O devices, sensors & communication modules	2
ETM314C.3	Remotely monitor data and control devices	5
ETM314C.4	Develop real life IoT based projects.	6

BCS-301: Web Design-II

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Understand the basic concept of PHP.
- How to use PHP to add some dynamic aspects to web pages.
- To learn how to download jQuery library and refer it to the Html page.
- Understanding Basic concept of AngularJS
- To explore AngularJS Components.

Unit I: **L: 10**

PHP: Introduction, PHP functions, PHP programming concepts, Handling Html Form with PHP, Working with file and Directories, Exception handling.

Unit II: **L: 10**

AngularJS : What is AngularJS, Difference between JavaScript and AngularJS, Advantages of Angular, AngularJS MVC architecture, Introduction to SPA(single page application), Setting up the environment, First App using MVC architecture, All about Angular expression , Number and String Expressions, Object Binding and Expressions, Working with Arrays, How to use expressions Angular vs JavaScript.

Unit III **L:10**

jQuery : JQuery introduction, Install and use jQuery library, Structure of JQuery, jQuery syntax, jQuery basics, First jQuery example, How to escape a special character, Basic selectors, JQuery DOM attributes, Traversal functions.

REFERENCE BOOKS:

1. Learning PHP, MySQL, books by 'O' riley Press Learning PHP, MySQL, books by ' O' riley Press.
2. Holzner Steven, PHP: The complete Reference, McGraw Hill education. Seventh edition, 2017.
3. Maximillian Schwarzmuller, Angular: The complete Guide, Packt Publishing.
4. Jonathan Chaffer, Karl Swedberg, jQuery Reference Guide, Packt Publishing.
5. David Flanagan, jQuery Pocket Reference, OReilly Media.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS316C.1	Design and implement web pages.	6
BCS316C.2	Design web forms and apply server-side validation.	6
BCS316C.3	Test and debug the PHP applications.	3
BCS316C.4	Build an awesome User Interface	3
BCS316C.5	Design to simplify the client-side scripting of HTML	6

BCS-302: Lab on Web Design-II

Course Objectives:

The objectives of the course are-

- Strong practice in JavaScript and PHP programming.
- Develop structurally well-formed web applications using AngularJS.
- Design and build rich interactive web applications using jQuery..

---

1. Develop a PHP program to demonstrate,
    - PHP functions
    - PHP programming concepts
2. Develop a PHP program to demonstrate,
    - Handling Html Form with PHP
    - Working with file and Directories
    - Exception handling
3. Write a PHP program to demonstrate the basic database operations.
4. Using AngularJS to create SPA.
5. Demonstrate all types of Expressions used in AngularJS.
6. Demonstrate the use of arrays in AngularJS.
7. Write AngularJS application to demonstrate object binding.
8. Write a jQuery code to check whether jQuery is loaded or not.
9. Write a jQuery code to demonstrate the jQuery selectors.
10. Write a jQuery code to demonstrate Traversal functions.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS316C.1	Design and implement a web application for a given problem-domain.	6
BCS316C.2	Design and implement a web application with Database design and structure using PHP.	6
BCS316C.3	Design and implement a web application using AngularJS application environment.	6
BCS316C.4	Design and manipulate a web page using jQuery.	6

Semester-VI

BDS 305: Machine Learning

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- The course aims to introduce the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Unit I: Introduction **L: 07**

Basic definitions, Hypothesis space and inductive bias, Bayes optimal classifier and Bayes error, Occam's razor, Curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit II: Regression **L: 08**

Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit III: Classification **L: 08**

Decision trees, Naive Bayes classifier, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine(SVM), Kernel functions.

Unit IV: Clustering **L: 07**

Approaches for clustering, distance metrics, K-means clustering, expectation maximization, hierarchical clustering, performance evaluation metrics, validation methods.

REFERENCE BOOKS:

1. Flach, P. (2015). Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press.
2. Mitchell, T.M. (2017). Machine Learning. McGraw Hill Education.
3. Christopher & Bishop, M. (2016). Pattern Recognition and Machine Learning. New York: Springer-Verlag.
4. Haykins, S.O. (2010). Neural Networks and Learning Machines. 3rd edition. PHI.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS321C.1	Differentiate between supervised and unsupervised learning tasks.	4
BDS321C.2	Differentiate between linear and non-linear classifiers.	4
BDS321C.3	Describe theoretical basis of SVM.	5
BDS321C.4	Implement various machine learning algorithms learnt in the course.	6

BDS 306: Data Visualization

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- This course introduces the basic concepts and techniques of Data visualization.
- The course aims to know the basics of data visualization, tables and visualizing data programmatically.
- To understand D3.js and advanced data visualization.

Unit I: Introduction to Data Visualization: L: 05

Acquiring and Visualizing Data, Simultaneous acquisition and visualization, Applications of Data Visualization, Keys factors of Data Visualization(Control of Presentation, Faster and Better JavaScript processing, Rise ofHTML5, Lowering the implementation Bar) Exploring the Visual Data Spectrum: charting Primitives (Data Points, Line Charts, Bar Charts, PieCharts, Area Charts), Exploring advanced Visualizations (Candlestick Charts, Bubble Charts, Surface Charts, Map Charts, Infographics). Making use ofHTML5 CANVAS, Integrating SVG.

Unit II: Basics of Data Visualization – Tables L: 07

Reading Data from Standard text files (.txt, .csv, XML), Displaying JSON content Outputting Basic Table Data (Building a table, Using Semantic Table, Configuring the columns), Assuring Maximum readability (Styling your table, Increasing readability, Adding dynamic Highlighting), Including computations, Using data tables library, relating data table to a chart.

Unit III: Visualizing data Programmatically L: 05

Creating HTML5 CANVAS Charts (HTML5 Canvas basics, Linear interpolations, A Simple Column Chart, Animations), Starting with Google charts (Google Charts API Basics, A Basic bar chart, A basic Pie chart, Working with Chart Animations).

Unit IV: Introduction to D3.js L: 05

Getting setup with D3, Making selections, changing selection's attribute, Loading and filtering External data : Building a graphic that uses all of the population distribution data, Data formats you can use with D3, Creating a server to upload your data, D3's function for loading data, Dealing with Asynchronous requests, Loading and formatting Large Data Sets.

Unit V:Advanced Data Visualization L: 08

Making charts interactive and Animated: Data joins, updates and exits, interactive buttons, Updating charts, Adding transactions, using keys

Adding a Play Button:

wrapping the update phase in a function, Adding a Play button to the page, Making the Play button go, Allow the user to interrupt the play, sequence

---

REFERENCE BOOKS:

1. Jon Raasch, Graham Murray, Vadim Ogievetsky, Joseph Lowery, “JavaScript and jQuery for Data Analysis and Visualization”, WROX
2. Ritchie S. King, Visual story telling with D3, Pearson.
3. Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment",O'Reilly, 2008.
4. A Julie Steele and Noah Iliinsky, Designing Data Visualizations: Representing Informational Relationships, O’Relly.
5. Andy Kirk, Data Visualization: A Successful Design Process, PAKT
6. Scott Murray, Interactive Data Visualization for Web, O’Relly
7. Nathan Yau, "Data Points: Visualization that means something", Wiley, 2013.
8. Tamara Munzner, Visualization Analysis and Design, AK Peters Visualization Series, CRC Press, Nov. 2014.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS322C.1	Explore various data visualization techniques in order to provide new insight.	5
BDS322C.2	Apply appropriate data visualization techniques to provide trends/insights for the given dataset.	3
BDS322C.3	Apply visualization tools / techniques for various data analysis tasks.	3
BDS322C.4	Given the application context for given data set, Design the information Dashboard for access information based on user criteria.	6

Or

BDS 306: Data Visualization

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- The course aims to introduce the basic concepts and techniques of Data Visualization so that a student can apply Data Visualization to a problem at hand.

Unit I **L: 08**

Value of Visualization – What is Visualization and Why do it: External representation – Interactivity – Difficulty in Validation. Data Abstraction: Dataset types – Attribute types – Semantics. Task Abstraction – Analyze, Produce, Search, Query. Four levels of validation – Validation approaches – Validation examples. Marks and Channels.

Unit II: **L: 10**

Rules of thumb – Arrange tables: Categorical regions – Spatial axis orientation – Spatial layout density. Arrange spatial data: Geometry – Scalar fields – Vector fields – Tensor fields. Arrange networks and trees: Connections, Matrix views – Containment. Map color: Color theory, Color maps and other channels.

Unit III: **L: 12**

Manipulate view: Change view over time – Select elements – Changing viewpoint – Reducing attributes. Facet into multiple views: Juxtapose and Coordinate views – Partition into views – Static and Dynamic layers – Reduce items and attributes: Filter – Aggregate. Focus and context: Elide – Superimpose – Distort – Case studies.

REFERENCE BOOKS:

1. Jon Raasch, Graham Murray, Vadim Ogievetsky, Joseph Lowery, JavaScript and jQuery for Data Analysis and Visualization, WROX.
2. Ritchie S. King, Visual story telling with D3, Pearson.
3. Ben Fry, Visualizing data: Exploring and explaining data with the processing environment,O'Reilly, 2008.
4. A Julie Steele and Noah Iliinsky, Designing Data Visualizations: Representing Informational Relationships, O'Relly.
5. Andy Kirk, Data Visualization: A Successful Design Process, PAKT.
6. Scott Murray, Interactive Data Visualization for Web, O'Relly.
7. Nathan Yau, Data Points: Visualization that means something, Wiley, 2013.
8. Tamara Munzner, Visualization Analysis and Design, AK Peters Visualization Series, CRC Press, Nov. 2014.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS322C.1	Understand the key techniques and theory behind data visualization.	2
BDS322C.2	Use effectively the various visualization structures (like tables, spatial data, tree and network etc.)	3
BDS322C.3	Evaluate information visualization systems and other forms of visual presentation for their effectiveness	5
BDS322C.4	Design and build data visualization systems.	6

BDS 307: Python Programming

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- This course is designed to introduce the student to the basics of programming using Python. The course covers the topics essential for developing well documented modular programs using different instructions and built-in data structures available in Python.

Unit I: **L: 05**

Introduction to Programming using Python: Installing Python on Linux and Windows, Structure of a Python Program, Functions, Interpreter shell, Indentation. Identifiers and keywords, Literals, Strings, Standard Data types, Basic operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator, Bit wise operator).

Unit II: **L: 05**

Building blocks of Python: Standard libraries in Python, notion of class, object and method.

Unit III: **L: 05**

Creating Python Programs: Input and Output Statements, Control statements: -branching, looping, Exit function, break, continue and pass, mutable and immutable structures. Testing and debugging a program.

Unit IV: **L: 05**

Built-in data structures: Strings, lists, Sets, Tuples and Dictionary and associated operations. Basic searching and sorting methods using iteration and recursion.

Unit V: **L: 05**

Advance Function Topics: Anonymous Function Lambda, Mapping Functions over Sequences: map, Functional Programming Tools: filter and reduce, Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages.

Unit VI: **L: 05**

Exception Handling and File Handling: Reading and writing text and structured files, Errors and Exceptions.
GUI Programming using TKinter

REFERENCE BOOKS:

1. Downey, A.B., (2015), Think Python–How to think like a Computer Scientist, 3rd edition. O'Reilly Media.
2. Taneja, S. & Kumar, N., (2017), Python Programming- A Modular Approach. Pearson Education.
3. Brown, M. C. (2001). The Complete Reference: Python, McGraw Hill Education.
4. Dromey, R. G. (2006), How to Solve it by Computer, Pearson Education.
5. Guttag, J.V.(2016), Introduction to computation and programming using Python. MIT Press.
6. Liang, Y.D. (2013), Introduction to programming using Python. Pearson Education
7. R. Nageswara Rao(2016), Core Python Programming, Dreamtech Press, 2016, ISBN-13: 9789351199427.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS323C.1	Develop, document, and debug modular python programs to solve computational problems.	6

BDS323C.2	Select a suitable programming construct and data structure for a situation.	3
BDS323C.3	Use built-in strings, lists, sets, tuples and dictionary in applications.	3
BDS323C.4	Define classes and use them in applications.	1
BDS323C.5	Use files for I/O operations.	3

BDS-308:Lab on Python Programming

Course Objectives:

The objectives of the course are-

- This course is designed to introduce the student to the basics of programming using Python. The course covers the topics essential for developing well documented modular programs using different instructions and built-in data structures available in Python.

---

1. Installing python and setting up environment
2. Develop programs to understand the control structures of python.
3. Develop programs to learn different types of structures (list, dictionary, tuples) in python.
4. Develop programs to learn concept of functions scoping, recursion, and list mutability.
5. Develop programs to understand object-oriented programming using python.
6. Develop programs for data structure algorithms using python – searching, sorting and hash tables.
7. Demonstrate the concept of exception handling using try/except/else Statement, Unified try/except/finally, try/finally Statement, raise Statement, assert Statement, catch multiple specific exceptions.
8. Demonstrate implementation of the Anonymous Function Lambda.
9. Demonstrate implementation functional programming tools such as filter and reduce.
10. Demonstrate the Module Creation, Module usage.
11. Develop programs to learn GUI programming using Tkinter

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS324C.1	Develop, document, and debug modular python programs to solve computational problems.	6
BDS324C.2	Select a suitable programming construct and data structure for a situation.	3
BDS324C.3	Use built-in strings, lists, sets, tuples and dictionary in applications.	3
BDS324C.4	Define classes and use them in applications.	1
BDS324C.5	Use files for I/O operations.	3

BDS-309:Lab on Data Visualization

Course Objectives:

The objectives of the course are-

- The course aims to introduce the basic concepts and techniques of Data Visualization so that a student can apply Data Visualization to a problem at hand.

---

1. Setup Environment for All the Tools
2. Develop the following Program Using HTML5 CANVAS and SVG TAG
   a. Develop the Different basic Graphical Shapes using HTM5 CANVAS
   b. Develop the Different Advanced Graphical Shapes using HTM5 CANVAS
   c. Develop the Different basic Graphical Shapes using HTM5 SVG
   d. Develop the Different Advanced Graphical Shapes using HTM5 SVG
3. Develop Following Program Using HTML5 and JavaScript
   a. Develop the simple bar chart usingTML5 CANVAS.
   b. Read the data .txt file and draw Data Table
   c. Read the data .txt file and draw Simple Bar Chart
   d. Read the data .csv file and draw Data Table
   e. Read the data .csv file and draw Column Bar Chart
   f. Read the data XML file and draw Data Table
   g. Read the data XML file and draw Simple Chart
   h. Read JSON Data and draw Data Table
   i. Read JSON Data and draw Simple Chart
4. Develop Following Program Using HTML5 and D3.js and Canvas.js
   a. Showing the data as a column chart (simple)
   b. Showing the data as a stacked column chart
   c. Showing the Data as a column chart for four age group
   d. Showing the data as a Line chart (single, fewer and multiple lines)
   e. Showing the data as a Pie Chart (single and multiple pie)
   f. Showing the data as a Bar Chart (Simple and multiple)
5. Develop Following Program Using HTML5 and Google Chats API and Map API
   a. Using Google Charts API Basics draw charts like a Bar chart.
   b. Using Google Charts API Basics draw charts like a Line chart.
   c. Using Google Charts API Basics draw PieChart.
   d. Using Google Charts API Basics draw Donut Chart.
   e. Using Google Charts API Basics draw Candle Chart.
   f. Using Google Charts API Basics draw other types of Charts.
   g. Using Google API read JSON file and create Google Map.
6. Build interconnected Dashboard usin.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BDS325C.1	Explore various data visualization techniques in order to provide new insight.	5
BDS325C.2	Apply appropriate data visualization techniques to provide	3

	trends/insights for the given dataset.	
BDS325C.3	Apply visualization tools / techniques for various data analysis tasks.	3
BDS325C.4	Given the application context for given data set, Design the information Dashboard for access information based on user criteria.	6

BCS-303: Data Analytics with Python(elective)
(Swayam)
(noc23-cs08)
*ClockHours:***60**

BCS-304: NLP
(Swayam)
(noc23-cs45)
(T)(4)(elective)

MTM-201: Theory of Computation
*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine and their relationships with formal languages.
- Students will also learn about the limitations of computing machines.

Unit I: **L: 10**

Languages: Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

Unit II: **L: 10**

Regular Expressions and Finite Automata: Regular expressions, Deterministic finite automata(DFA), Non-Deterministic finite automata, non-deterministic finite automata with ϵ-moves, Equivalence of FA's, finite automata with output.

Unit III: **L: 10**

Regular Languages: Pumping lemma for regular grammar, applications of pumping lemma, properties of regular languages, the relationship between regular languages and finite automata, Kleene's Theorem.

UNIT IV: **L: 10**

Non-Regular Languages and Context Free Grammars: Context-Free Grammars (CFG), derivation, Parse tree, ambiguity of CFG, simplification of grammar, Normal forms (CNF and GNF).

UNIT V: **L: 10**

Context-Free Languages (CFL) and PDA: Deterministic and non-deterministic Pushdown Automata (PDA), pumping lemma for CFL, properties of CFL, relationship between CFG and PDA.

UNIT VI: **L: 10**

Turing Machines and Models of Computations: Turing machine as a model of computation, configuration of simple Turing machine, Church Turing Thesis, Universal Turing Machine, decidability, halting problem.

REFERENCE BOOKS:

1. Cohen, D. I. A. (2011). Introduction to Computer Theory. 2nd edition. Wiley India.
2. Lewis, H.R. & Papadimitriou, H. R. (2002). Elements of the Theory of Computation. 6thedition. Prentice Hall of India (PHI)
3. Gopalkrishnan, G.L. (2019) Automata and Computability: A programmer's perspective. CRC Press.
4. Linz, P. (2016). An Introduction to Formal Languages and Automata.6th edition. Jones and Bartlett Learning.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM327C.1	Design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.	6
MTM327C.2	Apply pumping lemma to prove that a language is non-regular/non-context-free.	3
MTM327C.3	Describe limitations of a computing machine.	5

ETM-203:Digital Image Processing

*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- This course introduces students to the fundamentals of digital image processing, and various image transforms, image restoration techniques, image compression and segmentation used in digital image processing.

Unit I: **L: 10**

Introduction: Digital Image Fundamentals: Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships between Pixels Types of images.

Unit II: **L: 10**

Spatial Domain Filtering: Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothening Spatial Filters: Low pass filters, Order Statistics filters; Sharpening Spatial Filters: Laplacian filter.

Unit III: **L: 10**

Filtering in Frequency Domain: The Discrete Fourier Transformation (DFT), Frequency Domain Filtering: Ideal and Butterworth Low pass and High pass filters, DCT Transform (1D, 2D).

Unit IV: **L: 10**

Image Restoration**: Image Degradation/Restoration Process, Noise models, Noise Restoration Filters**

Image Compression: Fundamentals of Image Compression, Huffman Coding, Run Length Coding, JPEG.

Unit V: **L: 10**

Morphological Image Processing: Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

Unit VI: **L: 10**

Image Segmentation: Point, Line and Edge Detection, Thresholding, Region Based Segmentation.

REFERENCE BOOKS:

1. Gonzalez, R. C., & Woods, R. E. (2017). Digital Image Processing. 4th edition. Pearson Education.
2. Jain, A. K. (1988). Fundamentals of Digital Image Processing. 1st edition Prentice Hall of India.
3. Castleman, K. R. (1995.). Digital Image Processing. 1st edition. Pearson Education 2. Gonzalez, R. C., Woods, R. E., & Eddins, S. (2004). Digital Image Processing using MATLAB. Pearson Education Inc.
4. Schalkoff, D. (1989). Image Processing and Computer Vision. 1st edition. John Wiley and Sons.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
ETM327C.1	Describe the roles of image processing systems in a variety of applications.	5
ETM327C.2	Write programs to read/write and manipulate images: enhancement, segmentation, and compression, spatial filtering.	6
ETM327C.3	Develop Fourier transform for image processing in frequency domain.	6
ETM327C.4	Evaluate the methodologies for image segmentation and restoration.	5

Detail Syllabus
Information Technology
(Semester V and VI)

Semester-V

BIT-301: Mobile Application Development

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Understand basics of mobile application development and get introduced Android platform and its architecture. To develop the ability to create dynamic web pages.
- To learn activity creation and Android UI designing.
- Learn about the features and installation of Flutter.

Unit I: **L:05**

Mobile Application Development: Introduction to handheld devices, Device Applications Vs Desktop and Web application, overview of application development platform: Android, Comparison of Android with other Mobile OS.

Unit II: **L:05**

Hello, Android and Installations: background, what is android and what is not?, Open Mobile Development Platform, Native Android Applications, Android SDK Features, Introducing the Development Framework, Installation, Emulator.

Unit-III: **L:10**

Creating Applications, activities, and User Interfaces: Introducing the Application Manifest. Using the Manifest Editor. The Android Application Life Cycle. Understanding Application Priority and Process States. Externalizing Resources. A Closer Look at Android Activities. Fundamental Android UI Design. Introducing Views. Introducing Layouts and fragments, Using Adapters, Creating New Views.

Unit IV: **L:10**

Flutter: Introduction, Installation, Creating Simple Application in Android Studio, Architecture Application, Introduction to Dart Programming, Widgets, Layouts, Gestures, State Management, Animation, Writing Android Specific Code

---

REFERENCE BOOKS:

1. Reto Meier. Professional Android Application Development, Wrox Publications ISBN: 978-0 470-34471-2.
2. Rick Rogers, John Lombardo, Zigurd Mednieks, G. Blake Meike. Android Application Development: Programming with the Google SDK. O'Reilly ISBN 10: 0596521472 / ISBN 13: 9780596521479.
3. Marco L. Napoli, Beginning Flutter ®A Hands On Guide To App Development, Copyright © 2020 by John Wiley & Sons, Inc., Indianapolis, Indiana
4. Auxiliary Resources: https: //developer.android.com/ index.html, https:// www. tutorialspoint.com / flutter/ index.htm

---

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT311C.1	Compare android with another smartphone OS and desktop OS.	4
BIT311C.2	Understand software stack of android OS.	2
BIT311C.3	Understand Activity lifecycle and UI management.	2
BIT311C.4	Build Android and Flutter application using simple widgets and layouts.	6

BIT 302: Artificial Intelligence

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- This course introduces the basic concepts and techniques of Artificial Intelligence (AI).
- The course aims to introduce intelligent agents and reasoning, heuristic search techniques, gameplaying, knowledge representation, reasoning with uncertain knowledge.

Unit I: **L: 05**

Introduction, Definition, Future of Artificial Intelligence, characteristics of Intelligent Agents, Typical Intelligent Agents, Problem Solving Approach to Typical AI problems.

Unit II: **L: 10**

Problem solving Methods, Search Strategies, Uninformed, Informed, Heuristics, Local Search Algorithms and Optimization Problems, Searching with Partial, Observations, Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search, Game Playing, Optimal Decisions in Games, Alpha, Beta Pruning, Stochastic Games.

Unit III: **L: 10**

Knowledge Representation First Order Predicate Logic, Prolog Programming, Unification, Forward Chaining, Backward Chaining, Resolution, Knowledge Representation, Ontological Engineering, Categories and Objects, Events, Mental, Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

Unit IV: **L: 05**

Software Agents Architecture for Intelligent Agents, Agent communication, Negotiation and Bargaining, Argumentation among Agents, Trust and Reputation in Multi-agent systems, Applications AI applications

---

REFERENCE BOOKS:

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach, Prentice Hall, Third Edition, 2009.

2. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, University of California at Berkeley, Pearson education, 2020.
3. I. Bratko, Prolog: Programming for Artificial Intelligence, Fourth Edition, Addison-Wesley Educational Publishers Inc., 2011.
4. M. Tim Jones, Artificial Intelligence: A Systems Approach (Computer Science),Jones and Bartlett Publishers, Inc.; First Edition, 2008.
5. Nils J. Nilsson, The Quest for Artificial Intelligence, Cambridge University Press, 2009.
6. Gerhard Weiss, Multi Agent Systems, Second Edition, MIT Press, 2013.
7. David L. Poole and Alan K. Mackworth, Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT312C.1	Identify problems that are amenable to solution by specific AI methods.	4
BIT312C.2	Represent knowledge in Prolog and write code for drawing inferences.	6
BIT312C.3	Identify appropriate AI technique for the problem at hand.	4
BIT312C.4	Compare strengths and weaknesses of different artificial Intelligence techniques.	4
BIT312C.5	Sensitive towards development of responsible Artificial Intelligence.	6

BIT-303: Software Engineering

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- The course introduces fundamental Software Engineering approaches and techniques for software development.
- The students also develop a case study using appropriate software mode.

Unit I: Introduction to Software Engineering **L: 05**

Software and Software Engineering, Evolution of Software, Software Characteristics, Software Applications, Software Myths, Software Process, Software Development Life Cycle (SDLC).

Unit II: Software Requirements Specification **L: 06**

Waterfall model, prototyping, interactive enhancement, spiral model, Role of Management in software development. Role of metrics and measurement, Problem analysis, requirement specification, validation, metrics, monitoring and control.

Unit III: Software Design **L: 06**

Problem partitioning, abstraction, top-down and bottom-up design, Structured approach. Functional versus object-oriented approach, design specification and verification metrics, monitoring and control.

Unit IV: Software Coding and Testing **L: 07**

Top-down and bottom-up, structured programming, information hiding, programming style, and internal documentation. Verification, Metrics, monitoring and control, Levels of testing functional testing, structural testing, test plane, testcases specification, reliability assessment.

Unit V:Software Project Management **L: 06**

Cost estimation, Project scheduling, Staffing, Software configuration management, Quality assurance, Project Monitoring, Risk management.

REFERENCE BOOKS:

1. Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition. New Age International Publishers.
2. Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach.8th edition. McGraw-Hill
3. Jalote, P. (2005). An Integrated Approach to Software Engineering. 3rd edition. Narosa Publishing House.
4. Schwaber, K. & Sutherland, J. (2016). The Definitive Guide to Scrum: The Rules of the Game.[https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf]
5. Sommerville. (2011). Software Engineering. 9th edition. Addison Wesley.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT313C.1	Analyze and model customer's requirements and model its software design.	4
BIT313C.2	Use suitable software model for the problem at hand.	3
BIT313C.3	Estimate cost and efforts required in building software.	5
BIT313C.4	Analyze and compute impact of various risks involved in software development.	4
BIT313C.5	Design and build test cases, and perform software testing	6

BIT-304:Lab on Mobile Application Development

Course Objectives:

The objectives of the course are-

- To gain knowledge of installing Android Studio and Cross Platform Integrated Development Environment.
- To learn designing of User Interface and Layouts for Android App, intents to broadcast data.
- Install and use Flutter Framework.
- Use Dart Language, build cross platform applications.
- Build a simple application using a single codebase.

---

1. Create “Hello World” application. That will display “Hello World” in the middle of the screen in the red color with white background with change in fonts & styles of text.
2. Create List with string taken from resource folder (res>>value folder). On changing list value change image.
3. Create android UI such that, one screen has radio button of the types of cars. On selecting any carname, next screen should show car details like: name, company name, images if available, show different colors in which it is available.
4. Create android application that will display toast (Message) on android life cycle stages.
5. Create the application that will change color of screen, based on selected option from the menu.
6. Creating Simple Application in Android Studio
7. Create a simple application to demonstrate use of Flutter: Widgets, Layouts, Gestures, State Management, Animation
8. Build a simple interactive flutter app.
9. Building your own widget.
10. Build a simple quiz app in flutter for android.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT314C.1	Design and Implement User Interfaces and Layouts of Android App.	6
BIT314C.2	Use Intents for activity and broadcasting data in Android App.	3
BIT314C.3	Develop Flutter applications using Dart packages.	6

BCS-212: Introduction to Big Data Analytics (elective)
(Swayam)
(noc23-cs112)
*ClockHours:***60**

BCS-213:Computer Graphics (elective)
(Swayam)
(noc23-cs115)
*ClockHours:***60**

MTM-104:Linear Algebra
*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- The course provides mathematical material in Linear Algebra foundational for mathematics, engineering and the sciences.
- It focuses on linear equations, matrix methods, analytical geometry, and linear transformations.

Unit I: Systems of Linear Equations **L: 05**

System of Linear Equations, Augmented Matrix, Gaussian Elimination with Back-Substitution, Gauss-Jordan Elimination, Solving a Homogeneous System of Linear Equations.

Unit II: Matrices **L: 05**

Operations with Matrices: Addition, Scalar Multiplication, Matrix Multiplication, Identity, Transposes, Row operations, Finding the Inverse of a Matrix by Row Operations.

Unit III: Determinants **L: 10**

Determinant of 2x2 and 3x3 Matrices, Minors and Cofactors of a Square Matrix, Definition of the Determinant of a Square Matrix (Cofactor Theorem), Properties of Determinants, Cramer's Rule.

Unit IV: Vector Spaces **L: 10**

Vector Space, Subspace, Linear Combination, Linear Independence, Basis, Dimension, Finding a Basis of a Vector Space, Coordinates, Change of Basis.

Unit V:Inner Product Spaces **L: 10**

Inner Product, Length, Orthogonal Vectors, Triangle Inequality, Cauchy-Schwarz, Inequality, Orthonormal Orthogonal) Basis, Gram-Schmidt Process.

Unit VI: Linear Transformations **L: 10**

Linear Transformations and Matrices for Linear Transformation, Kernel and Range of a Linear Transformations, Change of Basis.

Unit VII: Eigenvalues and Eigenvectors **L: 10**

Eigenvalues and Eigenvectors, Definition of Eigenvalue and Eigenvector, Diagonalization, Symmetric Matrices and Orthogonal Diagonalization.

REFERENCE BOOKS:

1. Ron Larson, Calculus: Elementary Linear Algebra by, 8th edition, Cengage Learning,2017.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM314C.1	Employ techniques to classify and solve linear systems of equations.	3
MTM314C.2	Illustrate the use of matrices and determinants.	2
MTM314C.3	Utilize vector spaces and linear transformations.	3
MTM314C.4	Explore the concept of orthogonality in vector spaces.	4
MTM314C.5	Compute eigenvalues and eigenvectors of matrices.	5

ETM-202:Internet of Things

*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- The objective of this course is to impart necessary and practical knowledge of components of
- Internet of Things and develop skills required to build real-life IoT based projects.

Unit I: Introduction to IoT **L: 15**

Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing, Actuation, Basics of Networking, M2M and IoT Technology Fundamentals- Devices and gateways, Data management, Business processes in IoT, Everything as a Service(XaaS), Role of Cloud in IoT, Security aspects in IoT.

Unit II: Elements of IoT **L: 15**

Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces. Software Components- Programming API's (using Python/Node.js/Arduino) for Communication Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP.

Unit III: IoT Application Development **L: 15**

Solution framework for IoT applications- Implementation of Device integration, Data acquisition and integration, Device data storage- Unstructured data storage on cloud/local server, Authentication, authorization of devices.

Unit IV: IoT Case Studies **L: 15**

IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation.

REFERENCE BOOKS:

1. Vijay Madisetti, Arshdeep Bahga, Internet of Things, A Handson Approach, University Press.
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, Introduction to Internet of Things: A practical Approach, ETI Labs.
3. Pethuru Raj and Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press.
4. Jeeva Jose, Internet of Things, Khanna Publishing House, Delhi.
5. Adrian McEwen, Designing the Internet of Things, Wiley.
6. Raj Kamal, Internet of Things: Architecture and Design, McGraw Hill.
7. Cuno Pfister, Getting Started with the Internet of Things, O Reilly Media.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
ETM314C.1	Understand internet of Things and its hardware and software components	2
ETM314C.2	Interface I/O devices, sensors & communication modules	2
ETM314C.3	Remotely monitor data and control devices	6
ETM314C.4	Develop real life IoT based projects.	6

BCS-301: Web Design-II

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Understand the basic concept of PHP.
- How to use PHP to add some dynamic aspects to web pages.
- To learn how to download jQuery library and refer it to the Html page.
- Understanding Basic concept of AngularJS
- To explore AngularJS Components.

Unit I: **L: 10**

PHP: Introduction, PHP functions, PHP programming concepts, Handling Html Form with PHP, Working with file and Directories, Exception handling.

Unit II: **L: 10**

AngularJS : What is AngularJS, Difference between JavaScript and AngularJS, Advantages of Angular, AngularJS MVC architecture, Introduction to SPA(single page application), Setting up the environment, First App using MVC architecture, All about Angular expression , Number and String Expressions, Object Binding and Expressions, Working with Arrays, How to use expressions Angular vs JavaScript.

Unit III **L:10**

jQuery : JQuery introduction, Install and use jQuery library, Structure of JQuery, jQuery syntax, jQuery basics, First jQuery example, How to escape a special character, Basic selectors, JQuery DOM attributes, Traversal functions.

REFERENCE BOOKS:

1. Learning PHP, MySQL, books by 'O' riley Press Learning PHP, MySQL, books by ' O' riley Press.
2. Holzner Steven, PHP: The complete Reference, McGraw Hill education. Seventh edition, 2017.
3. Maximillian Schwarzmuller, Angular: The complete Guide, Packt Publishing.
4. Jonathan Chaffer, Karl Swedberg, jQuery Reference Guide, Packt Publishing.
5. David Flanagan, jQuery Pocket Reference, OReilly Media.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS316C.1	Design and implement web pages.	6
BCS316C.2	Design web forms and apply server-side validation.	6
BCS316C.3	Test and debug the PHP applications.	3
BCS316C.4	Build an awesome User Interface	3
BCS316C.5	Design to simplify the client-side scripting of HTML	6

BCS-302: Lab on Web Design-II

Course Objectives:

The objectives of the course are-

- Strong practice in JavaScript and PHP programming.
- Develop structurally well-formed web applications using AngularJS.
- Design and build rich interactive web applications using jQuery..

---

11. Develop a PHP program to demonstrate,
    - PHP functions
    - PHP programming concepts
12. Develop a PHP program to demonstrate,
    - Handling Html Form with PHP
    - Working with file and Directories
    - Exception handling
13. Write a PHP program to demonstrate the basic database operations.
14. Using AngularJS to create SPA.
15. Demonstrate all types of Expressions used in AngularJS.
16. Demonstrate the use of arrays in AngularJS.
17. Write AngularJS application to demonstrate object binding.
18. Write a jQuery code to check whether jQuery is loaded or not.
19. Write a jQuery code to demonstrate the jQuery selectors.
20. Write a jQuery code to demonstrate Traversal functions.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BCS316C.1	Design and implement a web application for a given problem-domain.	6
BCS316C.2	Design and implement a web application with Database design and structure using PHP.	6
BCS316C.3	Design and implement a web application using AngularJS application environment.	6
BCS316C.4	Design and manipulate a web page using jQuery.	6

Semester-VI

BIT-305: Machine Learning

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- The course aims to introduce the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

Unit I: Introduction **L: 07**

Basic definitions, Hypothesis space and inductive bias, Bayes optimal classifier and Bayes error, Occam's razor, Curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit II: Regression **L: 08**

Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit III: Classification **L: 08**

Decision trees, Naive Bayes classifier, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine(SVM), Kernel functions.

Unit IV: Clustering **L: 07**

Approaches for clustering, distance metrics, K-means clustering, expectation maximization, hierarchical clustering, performance evaluation metrics, validation methods.

REFERENCE BOOKS:

5. Flach, P. (2015). Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press.
6. Mitchell, T.M. (2017). Machine Learning. McGraw Hill Education.
7. Christopher & Bishop, M. (2016). Pattern Recognition and Machine Learning. New York: Springer-Verlag.
8. Haykins, S.O. (2010). Neural Networks and Learning Machines. 3rd edition. PHI.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT321C.1	Differentiate between supervised and unsupervised learning tasks.	4
BIT321C.2	Differentiate between linear and non-linear classifiers.	4
BIT321C.3	Describe theoretical basis of SVM	2
BIT321C.4	Implement various machine learning algorithms learnt in the course.	6

BIT-306: Web Design-III

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- Understand the JavaScript and technical concepts behind Node JS.
- Structure a Node application in modules.
- Build a Web Server in Node and understand how it really works.

Unit I: **L: 06**

Introduction to Node JS, Advantages, Traditional Web Server Model, Node.JS Process Model, Install Node.js on Windows, Working in REPL. Node JS Modules : Functions: Buffer, Module, Module Types, Core Modules, Local Modules, Module Exports.

Unit II: **L: 09**

Node Package Manager: Introduction, Installing packages Locally, Adding dependency in package.json, Installing packages globally, Updating packages. Web Server: Creating web server, Handling http requests, sending requests. File system: fs.readFile, Writing a file, opening a file, Deleting a file, Other I/O operations. Database Connectivity**.**

Unit III: **L: 08**

REACT JS : Introduction, Advantages of React JS, Work flow of React JS, Scope of React JS, overview of jsx, React Components: Properties, Setting Properties, Component Lifecycle, Updating Components, Writing React.js component, Mounting Components.

Unit IV: **L: 07**

JXS : Characteristics, Coding JSX, Expressions & Attributes, JSX Basics, Name spaced Components, Rendering HTML, Rendering React Components. Event handling in React.

REFERENCE BOOKS:

1. Valentin Bojinov, David Herron, Diogo Resende, Node.js Complete Reference Guide, 9781789952117, 1789952115, 2018, Packt Publishing.
2. Greg Lim, Beginning React, Kindly Edition.
3. Ray Yao, React.Js Programming, In 8 Hours, For Beginners, Learn Coding Fast: React.Js Language, Kindle Edition.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT322C.1	Design and develop React applications with JSX and state management.	6
BIT322C.2	Be familiar with Node.js application development and the React library.	2

BIT322C.3	Implement single page applications in React.	6

BIT-307: Python Programming

*ClockHours:***30**

Course Objectives:

The objectives of the course are-

- This course is designed to introduce the student to the basics of programming using Python.
- The course covers the topics essential for developing well documented modular programs using different instructions and built-in data structures available in Python.

Unit I: **L: 05**

Introduction to Programming using Python: Structure of a Python Program, Functions, Interpreter shell, Indentation. Identifiers and keywords, Literals, Strings, Basic operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator, Bit wise operator).

Unit II: **L: 05**

Building blocks of Python: Standard libraries in Python, notion of class, object and method.

Unit III: **L: 05**

Creating Python Programs: Input and Output Statements, Control statements: -branching, looping, Exit function, break, continue and pass, mutable and immutable structures. Testing and debugging a program.

Unit IV: **L: 05**

Built-in data structures: Strings, lists, Sets, Tuples and Dictionary and associated operations. Basic searching and sorting methods using iteration and recursion.

Unit V: **L: 05**

Advance Function Topics: Anonymous Function Lambda, Mapping Functions over Sequences: map, Functional Programming Tools: filter and reduce, Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages.

Unit VI: **L: 05**

Exception Handling and File Handling: Reading and writing text and structured files, Errors and Exceptions.
GUI Programming using TKinter

REFERENCE BOOKS:

1. Downey, A.B., (2015), Think Python–How to think like a Computer Scientist, 3rd edition. O'Reilly Media.
2. Taneja, S. & Kumar, N., (2017), Python Programming- A Modular Approach. Pearson Education.
3. Brown, M. C. (2001). The Complete Reference: Python, McGraw Hill Education.
4. Dromey, R. G. (2006), How to Solve it by Computer, Pearson Education.
5. Guttag, J.V.(2016), Introduction to computation and programming using Python. MIT Press.

6. Liang, Y.D. (2013), Introduction to programming using Python. Pearson Education

---

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT323C.1	Develop, document, and debug modular python programs to solve computational problems.	6
BIT323C.2	Select a suitable programming construct and data structure for a situation.	3
BIT323C.3	Use built-in strings, lists, sets, tuples and dictionary in applications.	3
BIT323C.4	Define classes and use them in applications.	1
BIT323C.5	Use files for I/O operations.	3

BIT-308:Lab on Python Programming

Course Objectives:

The objectives of the course are-

- This course is designed to introduce the student to the basics of programming using Python. The course covers the topics essential for developing well documented modular programs using different instructions and built-in data structures available in Python.

---

1. Installation of python environment
2. Develop programs to understand the control structures of python.
3. Develop programs to learn different types of structures (list, dictionary, tuples) in python.
4. Develop programs to learn concept of functions scoping, recursion and list mutability.
5. Develop programs to understand object oriented programming using python.
6. Develop programs for data structure algorithms using python – searching, sorting and hash tables.
7. Demonstrate the concept of exception handling using try/except/else Statement, Unified try/except/finally, try/finally Statement, raise Statement, assert Statement, catch multiple specific exceptions.
8. Demonstrate implementation of the Anonymous Function Lambda.
9. Demonstrate implementation functional programming tools such as filter and reduce.
10. Demonstrate the Module Creation, Module usage.
11. Develop programs to learn GUI programming using Tkinter.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT324C.1	Develop, document, and debug modular python programs to solve computational problems.	6
BIT324C.2	Select a suitable programming construct and data structure for a situation.	1
BIT324C.3	Use built-in strings, lists, sets, tuples and dictionary in applications.	3

BIT324C.4	Define classes and use them in applications.	1
BIT324C.5	Use files for I/O operations.	3

BIT-309:Lab on Web Design-III

Course Objectives:

The objectives of the course are-

- Learn why server-side JavaScript is useful using Node.js.
- Build an HTTP server using the core modules in Node.js.
- Use various React features including components and forms.

---

1. Create a Node.js file that will convert the output "Hello World!".
2. Create a Node.js Application that uses user defined module to return the factorial of given number.
3. Create a Node.js Application that uses user defined module circle.js which exports functions area() and circumference() and display details on console.
4. Create a Node.js Application that performs following operations on buffer data.
5. Create a Node.js file that demonstrates how to create database student DB and student table (Rno, Sname, Percentage) in MySQL.
6. Create a Node.js application to count no of lines in a file in file and display count on console.
7. Construct a reusable component using ReactJS.
8. Create a component using React JS.
9. Create an application to build interactive form using React JS event handling.
10. Create an application using React JS to fetch data from an API.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
BIT325C.1	Create server-side of web application using Node.js.	6
BIT325C.2	Create “classical” web applications with Node.js.	6
BIT325C.3	Implement a functional front-end web application using React	6
BIT325C.4	Use various React features including components and forms	3

BCS-305: Ethical Hacking (Swayam) (elective)

(noc-23-cs75)

T(4)

BCS-306: Secure Computations: Part 1 (Swayam) (elective)

(noc-23-cs75)

T(4)

MTM-201: Theory of Computation

*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine and their relationships with formal languages.
- Students will also learn about the limitations of computing machines.

Unit I: **L: 10**

Languages: Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

Unit II: **L: 10**

Regular Expressions and Finite Automata: Regular expressions, Deterministic finite automata(DFA), Non-Deterministic finite automata, non-deterministic finite automata with ϵ-moves, Equivalence of FA's, Finite automata with output.

Unit III: **L: 10**

Regular Languages: Pumping lemma for regular grammar, applications of pumping lemma, properties of regular languages, the relationship between regular languages and finite automata, Kleene's Theorem.

Unit IV: **L: 10**

Non-Regular Languages and Context Free Grammars: Context-Free Grammars (CFG), derivation, Parse tree, ambiguity of CFG, simplification of grammar, Normal forms (CNF and GNF).

Unit V: **L: 10**

Context-Free Languages (CFL) and PDA: Deterministic and non-deterministic Pushdown Automata (PDA), pumping lemma for CFL, properties of CFL, relationship between CFG and PDA.

Unit VI: **L: 10**

Turing Machines and Models of Computations: Turing machine as a model of computation, configuration of simple Turing machine, Church Turing Thesis, Universal Turing Machine, decidability, halting problem.

REFERENCE BOOKS:

1. Cohen, D. I. A. (2011). Introduction to Computer Theory. 2nd edition. Wiley India.
2. Lewis, H.R. & Papadimitriou, H. R. (2002). Elements of the Theory of Computation. 6thEdition. Prentice Hall of India (PHI)
3. Gopal Krishnan, G.L. (2019) Automata and Computability: A programmer's perspective. CRC Press.
4. Linz, P. (2016). An Introduction to Formal Languages and Automata.6th edition. Jones and Bartlett Learning.

Course Outcome:
After completion of this course students shall be able to-

CO No.	CO	Cognitive level
MTM327C.1	Design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.	6
MTM327C.2	Apply pumping lemma to prove that a language is non-regular/non-context-free.	3
MTM327C.3	Describe limitations of a computing machine.	2

ETM-203: Digital Image Processing

*ClockHours:***60**

Course Objectives:

The objectives of the course are-

- This course introduces students to the fundamentals of digital image processing, and various image transforms, image restoration techniques, image compression and segmentation used in digital image processing.

Unit I: **L: 10**

Introduction: Digital Image Fundamentals: Brightness, Adaptation and Discrimination, Lightand Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships between Pixels Types of images.

Unit II: **L: 10**

Spatial Domain Filtering: Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothening Spatial Filters: Low pass filters, Order Statistics filters; Sharpening Spatial Filters: Laplacian filter.

Unit III: **L: 10**

Filtering in Frequency Domain: The Discrete Fourier Transformation (DFT), Frequency Domain Filtering: Ideal and Butterworth Low pass and High pass filters, DCT Transform (1D,2D).

Unit IV: **L: 10**

Image Restoration**: Image Degradation/Restoration Process, Noise models, Noise Restoration Filters**

Image Compression: Fundamentals of Image Compression, Huffman Coding, Run Length Coding, JPEG.

Unit V: **L: 10**

Morphological Image Processing: Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

Unit VI: **L: 10**

Image Segmentation: Point, Line and Edge Detection, Thresholding, Region Based Segmentation**.**

REFERENCE BOOKS:

1. Gonzalez, R. C., & Woods, R. E. (2017). Digital Image Processing. 4th edition. Pearson Education.
2. Jain, A. K. (1988). Fundamentals of Digital Image Processing. 1st edition Prentice Hall of India.
3. Castleman, K. R. (1995.). Digital Image Processing. 1st edition. Pearson Education2. Gonzalez, R. C., Woods, R. E., & Eddins, S. (2004). Digital Image Processing using MATLAB. Pearson Education Inc.
4. Schalkoff, D. (1989). Image Processing and Computer Vision. 1st edition. John Wiley and Sons.

Course Outcome:

After completion of this course students shall be able to-

CO No.	CO	Cognitive level
ETM327C.1	Describe the roles of image processing systems in a variety of applications;	2
ETM327C.2	Write programs to read/write and manipulate images: enhancement, segmentation, and compression, spatial filtering.	6
ETM327C.3	Develop Fourier transform for image processing in frequency domain.	6
ETM327C.4	Evaluate the methodologies for image segmentation, restoration.	5