Subject	UT-1	Half-Yearly Exam			Annual Exam	
			Project	UT-2		Project
ications oxix 2.0)	Chapter-1: Introduction to Object Oriented Programming concepts (i) Principles of Object Oriented Programming, All the four principles of Object Oriented Programming should be defined and explained using real life examples (Data abstraction, Inheritance, Polymorphism, Encapsulation). (ii) Introduction to JAVA - Types of java programs, Java Compilation process, Java Source code, Byte code, Object code, (JVM), Features of JAVA, steps involved in compilation process, definitions of source code,	concepts (i) Principles of Object Oriented Programming, All the four principles of Object Oriented	Write any 10 Program in JAVA which are based on the half-yearly syllabus	Chapter-6: Mathematical Library Methods Introduction to package java.lang [ default ], methods of Math class. pow(x,y), sqrt(x), cbrt(x), ceil(x), floor(x), round (x), abs(a), max(a, b), min(a,b), random( ). Java expressions — using all the operators and methods of Math class.	Character set, ASCII code, Unicode, Escape sequences, Tokens, Constants and	Write any 10 programs in JAVA which are based on the annual syllabus

Chapter-2: Elementary Concept of	Chapter-2: Elementary Concept of	Chapter-7: Cond	ditional constructs	Chapter-4: Operators in Java	
Objects and Classes	Objects and Classes	in Java		Forms of operators, Types of	
		Application of if,	, if else, if else if	operators, Counters,	
Modelling entities and their	Modelling entities and their		ase, default, break.	Accumulators, Hierarchy of	
behaviour by objects, a class as a	behaviour by objects, a class as a			operators, 'new' operator, dot ( . )	
specification for objects and as an	specification for objects and as an		ement, fall through	operator. Forms of operators	
object factory, computation as	object factory, computation as	condition in swit		(Unary, Binary, Ternary), types	
message passing/method calls	message passing/method calls	driven programs	s, System.exit(0) - to	of operators (Arithmetic,	
between objects (many examples	between objects (many examples	terminate the pr	rogram.	Relational, Logical,	
should be done to illustrate this).	should be done to illustrate this).			Assignment, Increment,	
Objects encapsulate state	Objects encapsulate state			Decrement, Short hand	
(attributes) and have behaviour	(attributes) and have behaviour			operators), Discuss precedence and	
(methods). Class as a user defined	(methods). Class as a user defined			associativity of operators, prefix	
data type. A class may be regarded	data type. A class may be regarded			and postfix, Creation of dynamic	
as a blueprint to create objects. It	as a blueprint to create objects. It			memory by using new operator,	
may be viewed as a factory that	may be viewed as a factory that			invoking members	
produces similar objects.	produces similar objects.			of class using dot operator,	
				Introduce System.out.println() and	
				System.out.print() for simple	
				output. (Bitwise and shift operators	
				are not included).	
	Chapter-3: Values and data types			Chapter-5: Input in Java	
	enapter of values and data types			Chapter 3: input in sava	
	Character set, ASCII code, Unicode,			Initialization, Parameter,	
	Escape			introduction to packages,	
	sequences, Tokens, Constants and			Input streams (Scanner Class), types	
	Variables, Data			of errors, types of comments	
	types, type conversions.			Initialization – Data before	
	Escape sequences [\n, \t, \ \", \'],			execution, Parameters – at the time	
	Tokens and its			of execution, input stream – data	
	types [keywords, identifiers,			entry during execution – using	
	literals, punctuators,			methods of Scanner class	
	operators], primitive types and non-			[nextShort(), nextInt(), nextLong(),	
	primitive types			nextFloat (), nextDouble(), next(),	
	with examples, Introduce the			nextLine(), next ().charAt(0)]	
	primitive types with			Discuss different types of errors	
	size in bits and bytes, Implicit type			occurring during execution and	
	conversion and			compilation of the program (syntax	
	Explicit type conversion.			errors, runtime errors and logical	
				errors). Single line comment (//)	
				and multiline comment (/* */ )	
				, ,	

Charles A County	-		C. Mathematical Library
Chapter-4: Operators in Ja	va		-6: Mathematical Library
			s Introduction to package
Forms of operators, Types	of		g [ default ], methods of
operators, Counters,			ass. pow(x,y), sqrt(x),
Accumulators, Hierarchy of			ceil(x), floor(x), round (x),
operators, 'new' operator,	dot ( . )	abs(a), ma	nax(a, b), min(a,b), random(
operator. Forms of operato	ors	). Java exp	xpressions – using all the
(Unary, Binary, Ternary), ty	pes	operators	rs and methods of Math
of operators (Arithmetic,		class.	
Relational, Logical, Assignm	nent,		
Increment, Decrement, Sho	ort hand		
operators), Discuss precede			
associativity of operators, p			
and postfix, Creation of dyr			
memory by using new oper			
invoking members			
of class using dot operator,			
Introduce System.out.print			
System.out.print() for simp			
output. (Bitwise and shift o			
are not included).	perators		
are not included).			
Chapter-5: Input in Java		i i i i i i i i i i i i i i i i i i i	-7: Conditional constructs
		in Java	
Initialization, Parameter,		in Java Application	ion of if, if else, if else if
Initialization, Parameter, introduction to packages,		in Java Applicatio ladder, sv	ion of if, if else, if else if witch-case, default, break.
Initialization, Parameter,	ss), types	in Java Applicatio ladder, sv	ion of if, if else, if else if
Initialization, Parameter, introduction to packages,		in Java Application ladder, sw if, if else,	ion of if, if else, if else if witch-case, default, break.
Initialization, Parameter, introduction to packages, Input streams (Scanner Clas	ts	in Java Application ladder, sw if, if else, case, brea	ion of if, if else, if else if witch-case, default, break. , if else if, Nested if, switch
Initialization, Parameter, introduction to packages, Input streams (Scanner Clas of errors, types of commen	ts	in Java Application ladder, sw if, if else, case, breacondition	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through
Initialization, Parameter, introduction to packages, Input streams (Scanner Clar of errors, types of commen Initialization – Data before	the time	in Java Application ladder, sw if, if else, case, brea condition driven pre	ion of if, if else, if else if witch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commentalization – Data before execution, Parameters – at	ts the time – data	in Java Application ladder, sw if, if else, case, brea condition driven pre	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream	ts the time – data	in Java Application ladder, sw if, if else, case, brea condition driven pre	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commentalization – Data before execution, Parameters – at of execution, input stream entry during execution – us	the time – data sing	in Java Application ladder, sw if, if else, case, brea condition driven pre	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next	the time  – data sing tLong(),	in Java Application ladder, sw if, if else, case, brea condition driven pre	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), nextI	the time - data sing tLong(), , next(),	in Java Application ladder, sw if, if else, case, brea condition driven pre	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble(), nextLine(), next ().charAt()	the time - data sing tLong(), , next(),	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of execution into packages, Input streams of execution – us methods of Scanner class [nextShort(), nextInt(), nextDouble()]	the time - data sing tLong(), , next(), 0)] rrors	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of execution	the time - data sing  tLong(), , next(),  0)]  rrors and	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Clasof errors, types of commentalization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of e occurring during execution compilation of the program	the time - data sing  tLong(), , next(),  D)]  rrors and n (syntax	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of e occurring during execution compilation of the program errors, runtime errors and	the time - data sing  tLong(), , next(),  D)]  rrors and a (syntax logical	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of comment Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of execution of the program errors, runtime errors and errors).Single line commen	the time - data sing  tLong(), , next(),  D)]  rrors and n (syntax logical t (//) and	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of commen Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of e occurring during execution compilation of the program errors, runtime errors and	the time - data sing  tLong(), , next(),  D)]  rrors and n (syntax logical t (//) and	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of comment Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of execution of the program errors, runtime errors and errors).Single line commen	the time - data sing  tLong(), , next(),  D)]  rrors and n (syntax logical t (//) and	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to
Initialization, Parameter, introduction to packages, Input streams (Scanner Class of errors, types of comment Initialization – Data before execution, Parameters – at of execution, input stream entry during execution – us methods of Scanner class [nextShort(), nextInt(), next nextFloat (), nextDouble() nextLine(), next ().charAt() Discuss different types of execution of the program errors, runtime errors and errors).Single line commen	the time - data sing  tLong(), , next(),  D)]  rrors and n (syntax logical t (//) and	in Java Application ladder, sw if, if else, case, brea condition driven pro	ion of if, if else, if else if switch-case, default, break. , if else if, Nested if, switch eak statement, fall through n in switch case, Menu rograms, System.exit(0) - to

		Chapter-8. Iterative constructs in Java  Definition, Types of looping statements, entry controlled loops [ for, while], exit controlled loop [do while] , variations in looping statements, and Jump statements. Syntax of entry and exit controlled loops, break and continue, Simple programs illustrating all three loops, inter conversion from for — while — do while finite and infinite, delay, multiple counter variables (initializations and updations). Demonstrate break and continue statements with the	
		while – do while finite and infinite, delay, multiple counter variables (initializations and updations). Demonstrate break	