CMR ENGINEERING COLLEGE

ENGINEERING COLLEGE
EXPLORE TO INVENT

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE INSTRUCTOR NAME: MOHAMMED AZHAR ACADEMIC YEAR: 2024-25

SUBJECT NAME: OOPs THROUGH JAVA SECTION :A & F

EMAIL-ID: taufeeq.azhar@cmrec.ac.in CLASS ROOM NO: D-201&219

CONTACT NO: 8125671754

SEM- START DATE AND END DATE: 29-07-24 T0 29-11-24

CONTENTS OF COURSE FILE

15. Previous year question papers

16. Power point presentations (PPTs)

17. Innovative Teaching method

18. References (Textbook/Websites/Journals)

HOD

1. DEPARTMENT VISION & MISSION

VISION:

To produce globally competent and industry-ready graduates in Computer Science & Engineering by imparting quality education with the know-how of cutting-edge technology and holistic personality.

MISSION:

1. To offer high-quality education in Computer Science & Engineering in order to build core competence for the graduates by laying a solid foundation in Applied Mathematics and program frameworkwith a focus on concept building.

2. The department promotes excellence in teaching, research, and collaborative activities to prepare graduates for a professional career or higher studies.

3. Creating an intellectual environment for developing logical skills and problem-solving strategies, thus developing, an able and proficient computer engineer to compete in the current global scenario.

2. LIST OF PEOs, POs AND PSOs

2.1 Program Educational Objectives (PEO):

PEO 1: Excel in professional career and higher education by acquiring knowledge of mathematical computing and engineering principles.

PEO 2: To provide an intellectual environment for analyzing and designing computing systems for technical needs.

PEO 3: Exhibit professionalism to adapt current trends using lifelong learning with legal and ethical responsibilities.

PEO 4: To produce responsible graduates with effective communication skills and multidisciplinary practices to serve society and preserve the environment.

2.2. Program Outcomes (POs):

Engineering Graduates will be able to satisfy these NBA graduate attributes:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

8. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

9. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
10. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

11. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

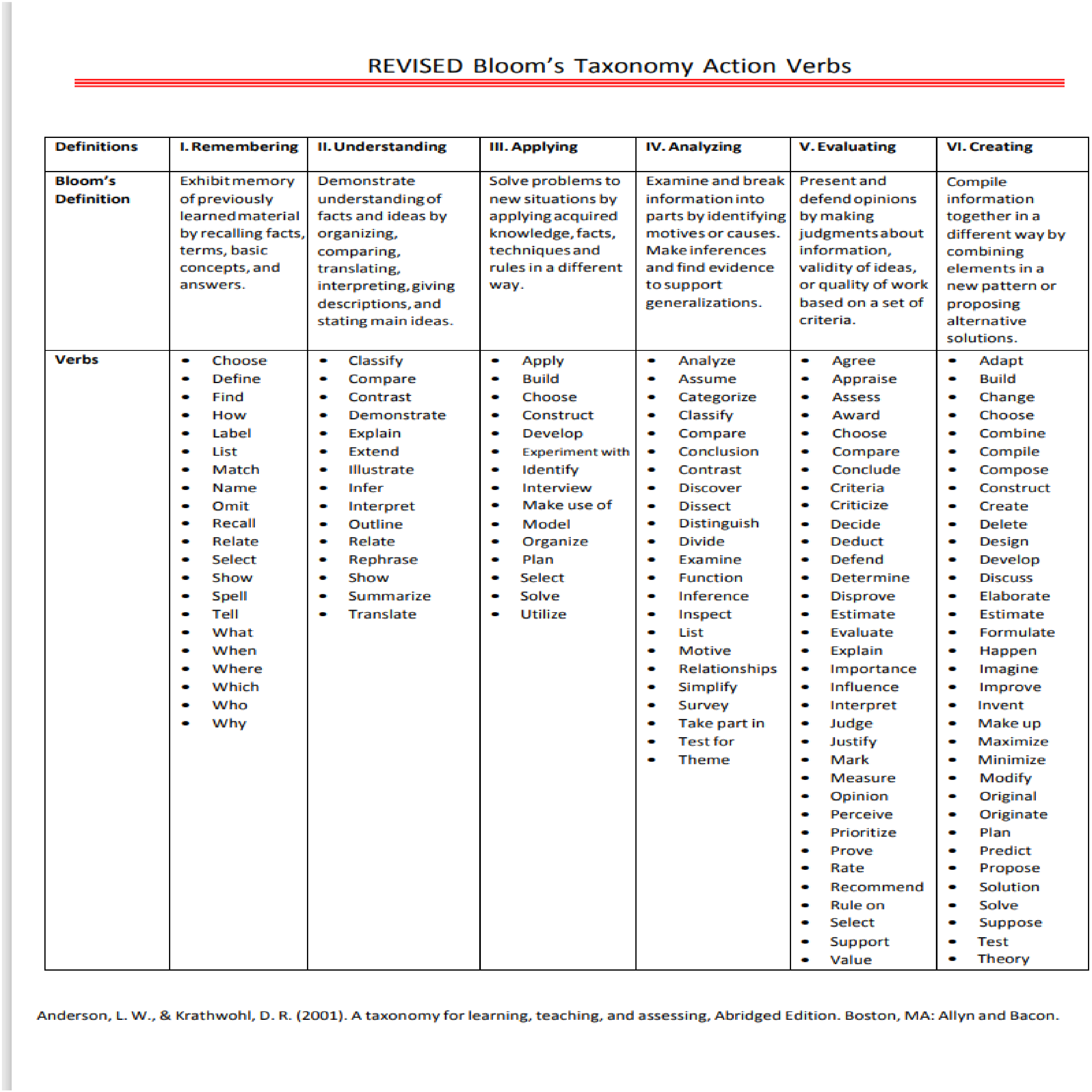
2.3 Program Specific Outcomes (PSOs):

PSO1: Professional Skills and Foundations of Software development: Ability to analyze, design and develop applications by adopting the dynamic nature of Software developments.
PSO2: Applications of Computing and Research Ability: Ability to use knowledge in cutting edge technologies in identifying research gaps and to render solutions with innovative ideas.

3. COURSE OUTCOMES

CO1	Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection **[BTL1,BTL2,BTL3][REMEMBERING,UNDERSTANDING,APPLYING]**
CO2	Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implements keywords. **[BTL2,BTL3][UNDERSTANDING,APPLYING]**
CO3	Use multithreading concepts to develop inter process communication. **[BTL2,BTL6][UNDERSTANDING,CREATING]**
CO4	Understand the process of graphical user interface design and implementation using AWT or swings..**[BTL2,BTL3,BTL6][UNDERSTANDING,APPLYING,CREATING]**
CO5	Develop applets that interact abundantly with the client environment and deploy on the server..**[BTL3,BTL6][APPLYING,CREATING]**
PSO1	**Professional Skills and Foundations of Software development**: Ability to analyze, design, and develop applications by adopting the dynamic nature of

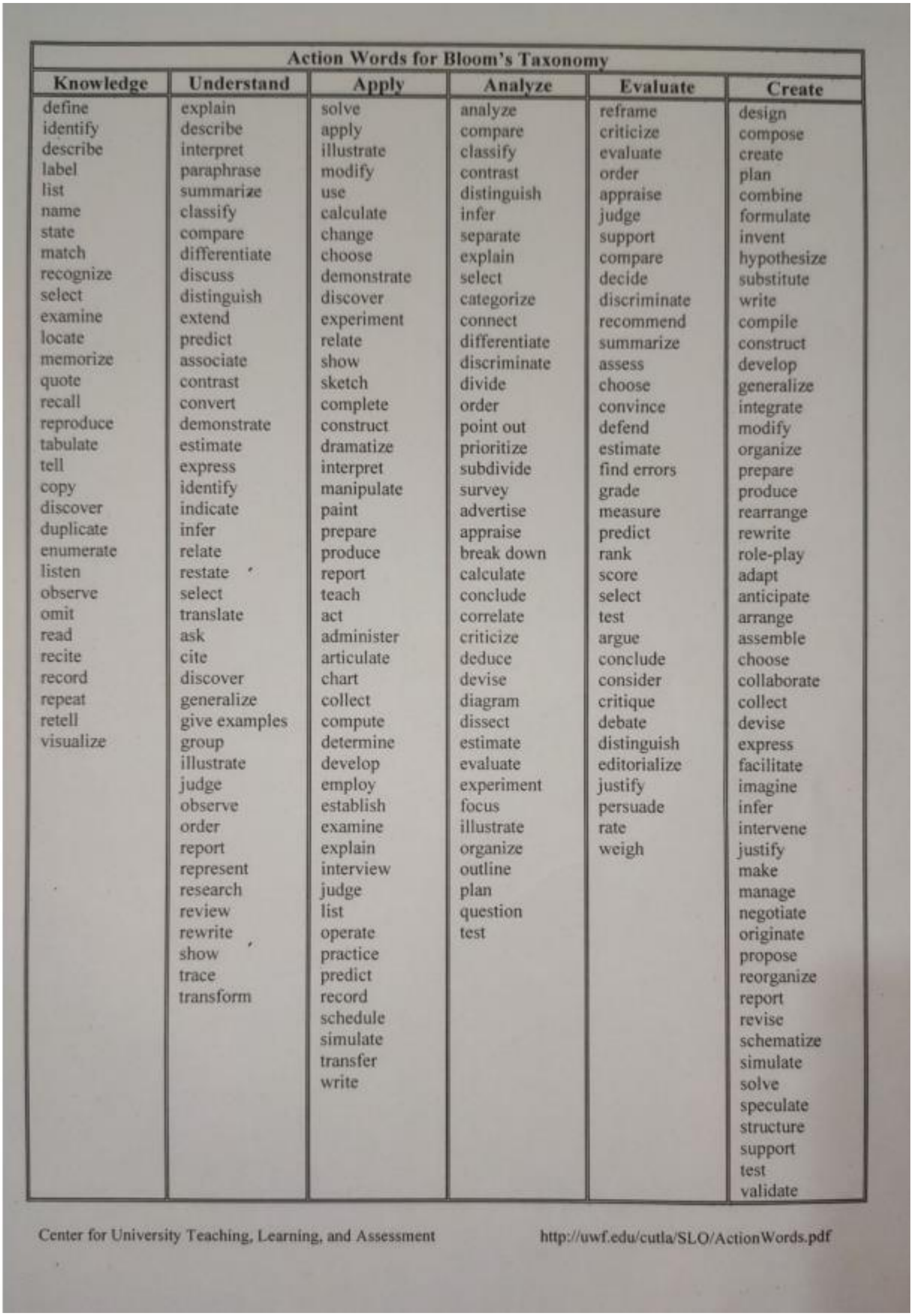
	Software development.
PSO2	**Applications of Computing and Research Ability**: Ability to use knowledge in cutting-edge technologies in identifying research gaps and to render cutting-edge innovative ideas.



REVISED Bloom's Taxonomy Action Verbs

Definitions	**I. Remembering**	**II. Understanding**	**III. Applying**	**IV. Analyzing**	**V. Evaluating**	**VI. Creating**
Bloom's Definition	Exhibit memory of previously learned material by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas.	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support generalizations.	Present and defend opinions by making judgments about information, validity of ideas, or quality of work based on a set of criteria.	Compile information together in a different way by combining elements in a new pattern or proposing alternative solutions.
Verbs	• Choose • Define • Find • How • Label • List • Match • Name • Omit • Recall • Relate • Select • Show • Spell • Tell • What • When • Where • Which • Who • Why	• Classify • Compare • Contrast • Demonstrate • Explain • Extend • Illustrate • Infer • Interpret • Outline • Relate • Rephrase • Show • Summarize • Translate	• Apply • Build • Choose • Construct • Develop • Experiment with • Identify • Interview • Make use of • Model • Organize • Plan • Select • Solve • Utilize	• Analyze • Assume • Categorize • Classify • Compare • Conclusion • Contrast • Discover • Dissect • Distinguish • Divide • Examine • Function • Inference • Inspect • List • Motive • Relationships • Simplify • Survey • Take part in • Test for • Theme	• Agree • Appraise • Assess • Award • Choose • Compare • Conclude • Criteria • Criticize • Decide • Deduct • Defend • Determine • Disprove • Estimate • Evaluate • Explain • Importance • Influence • Interpret • Judge • Justify • Mark • Measure • Opinion • Perceive • Prioritize • Prove • Rate • Recommend • Rule on • Select • Support • Value	• Adapt • Build • Change • Choose • Combine • Compile • Compose • Construct • Create • Delete • Design • Develop • Discuss • Elaborate • Estimate • Formulate • Happen • Imagine • Improve • Invent • Make up • Maximize • Minimize • Modify • Original • Originate • Plan • Predict • Propose • Solution • Solve • Suppose • Test • Theory

Anderson, L. W., & Krathwohl, D. R. (2001). A taxonomy for learning, teaching, and assessing, Abridged Edition. Boston, MA: Allyn and Bacon.



Action Words for Bloom's Taxonomy

Knowledge	Understand	Apply	Analyze	Evaluate	Create
define	explain	solve	analyze	reframe	design
identify	describe	apply	compare	criticize	compose
describe	interpret	illustrate	classify	evaluate	create
label	paraphrase	modify	contrast	order	plan
list	summarize	use	distinguish	appraise	combine
name	classify	calculate	infer	judge	formulate
state	compare	change	separate	support	invent
match	differentiate	choose	explain	compare	hypothesize
recognize	discuss	demonstrate	select	decide	substitute
select	distinguish	discover	categorize	discriminate	write
examine	extend	experiment	connect	recommend	compile
locate	predict	relate	differentiate	summarize	construct
memorize	associate	show	discriminate	assess	develop
quote	contrast	sketch	divide	choose	generalize
recall	convert	complete	order	convince	integrate
reproduce	demonstrate	construct	point out	defend	modify
tabulate	estimate	dramatize	prioritize	estimate	organize
tell	express	interpret	subdivide	find errors	prepare
copy	identify	manipulate	survey	grade	produce
discover	indicate	paint	advertise	measure	rearrange
duplicate	infer	prepare	appraise	predict	rewrite
enumerate	relate	produce	break down	rank	role-play
listen	restate	report	calculate	score	adapt
observe	select	teach	conclude	select	anticipate
omit	translate	act	correlate	test	arrange
read	ask	administer	criticize	argue	assemble
recite	cite	articulate	deduce	conclude	choose
record	discover	chart	devise	consider	collaborate
repeat	generalize	collect	diagram	critique	collect
retell	give examples	compute	dissect	debate	devise
visualize	group	determine	estimate	distinguish	express
	illustrate	develop	evaluate	editorialize	facilitate
	judge	employ	experiment	justify	imagine
	observe	establish	focus	persuade	infer
	order	examine	illustrate	rate	intervene
	report	explain	organize	weigh	justify
	represent	interview	outline		make
	research	judge	plan		manage
	review	list	question		negotiate
	rewrite	operate	test		originate
	show	practice			propose
	trace	predict			reorganize
	transform	record			report
		schedule			revise
		simulate			schematize
		transfer			simulate
		write			solve
					speculate
					structure
					support
					test
					validate

Center for University Teaching, Learning, and Assessment http://uwf.edu/cutla/SLO/ActionWords.pdf

4. SYLLABUS COPY

UNIT - I

Object oriented thinking and Java Basics- Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. A way of viewing world – Agents, responsibility, messages, methods, History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT – II

Inheritance, Packages and Interfaces – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT – III

Exception handling and Multithreading-- Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, Exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT – IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, checkbox groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

UNIT - V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

TEXT BOOKS:

1. Java the complete reference, 7th edition, Herbert schildt, TMH.

2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley & sons.

2. An Introduction to OOP, third edition, T. Budd, Pearson education.

3. Introduction to Java programming, Y. Daniel Liang, Pearson education.

4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson. 5. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.

6. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education 7. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH. Java and Object Orientation, an introduction, John Hunt, second edition, Springer. 9. Maurach's Beginning Java2 JDK 5, SPD.

5. INDIVIDUAL TIME TABLE (MOHAMMED AZHAR)

	I	II	III	IV		V	VI	VII
MON	OOPs- JAVA-A	OOPs- JAVA-F	JAVA LAB-A		LUNCH		OOPs- JAVA-F	OOPs- JAVA-F
TUE				OOPs- JAVA-A		OOPs- JAVA-F		
WED	OOPs- JAVA-F			OOPs- JAVA-A				
THU			OOPs- JAVA-A	OOPs- JAVA-F			OOPs- JAVA-A	
FRI		OOPs- JAVA-A						
SAT	JAVA LAB-F							

6. SESSION PLAN/LESSON PLAN

S.NO	Topic (JNTU syllabus)	Sub-Topic	NO. OF LECTURES REQUIRED	Suggested Books	Teaching Methods
UNIT – I					
1	**Object oriented thinking and Java Basics**	Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms.	1	T1, T2	M1,M4
2		A way of viewing world – Agents, responsibility, messages, methods,		T1, T2	M1,M4
3		History of Java, Java buzzwords,	1	T1, T2	M1,M4
4		data types, variables, scope and lifetime of variables		T1, T2	M1,M5
5		arrays, operators, expressions, control statements,	1	T1, T2	M1,M5
6		type conversion and casting, simple java program	1	T1, T2	M1,M5
7		concepts of classes, objects		T1, T2	M1,M4

8		constructors, methods	1	T1, T2	M1,M5
9		access control, this keyword	1	T1, T2	M1,M5
10		garbage collection, overloading methods and constructors	1	T1, T2	M1,M5
11		method binding, inheritance	1	T1, T2	M1,M5
12		overriding and exceptions	1	T1, T2	M1,M5
13		parameter passing, recursion	1	T1, T2	M1,M5
14		nested and inner classes	1	T1, T2	M1,M5
15		exploring string class	1	T1, T2	M1,M5
16		**UNIT-I**	**L15**	**SLIP TEST**	
UNIT – II					
17	**Inheritance, Packages and Interfaces**	Hierarchical abstractions, Base class object, subclass, subtype, substitutability	**1**	T1, T2	M1.M4
18		forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance	**1**	T1, T2	M1,M4

19		Member access rules, super uses, using final with inheritance	**1**	T1, T2	M1,M5
20		polymorphism- method overriding,.	**1**	T1, T2	M1,M5
21		abstract classes, the Object class	**1**	T1, T2	M1,M5
22		Defining, Creating and Accessing a Package,	**1**	T1, T2	M1,M5
23		Understanding CLASSPATH, importing packages	**1**	T1, T2	M1,M5
24		differences between classes and interfaces	**1**	T1, T2	M4,M5
25		defining an interface, implementing interface, applying interfaces,	**1**	T1, T2	M1,M5
26		variables in interface and extending interfaces	**1**	T1, T2	M1,M5
27		Exploring string class.	**1**	T1, T2	M1,M5
28		**UNIT-II**	**L26**	**SLIP TEST**	
UNIT-III					
29	**Exception handling and Multithreading**	Concepts of exception handling, benefits of exception handling	1	T1, T2	M1,M4
30		Termination or resumptive models	**1**	T1, T2	M1,M4
31		exception hierarchy, usage of try, catch, throw, throws and finally	**1**	T1, T2	M1,M4,M5
32		built in exceptions, creating own	**1**	T1, T2	M1,M5

		exception subclasses			
33		String handling	**1**	T1, T2	M1,M5
34		Exploring java.util.	**1**	T1, T2	M1,M5
35		Differences between multithreading and multitasking, thread life cycle	**1**	T1, T2	M4
36		creating threads, thread priorities, synchronizing threads	**1**	T1, T2	M1
37		inter thread communication, thread groups, daemon threads.	**1**	T1, T2	M1,M5
38		Enumerations	**1**	T1, T2	M1,M5
39		autoboxing, annotations,	**1**	T1, T2	M1,M5
40		generics	**1**	T1, T2	M1,M5
41		**UNIT-III**	**L38**	**SLIP TEST**	
42		**UNIT-III**	**SLIP TEST**		
UNIT-IV					
43	**Event Handling**	Events, Event sources, Event classes, Event Listeners	**1**	T1, T2	M4
44		Delegation event model, handling mouse and keyboard events	**1**	T1,T2	M1,M4

45		Adapter classes.	**1**	T1, T2	M1,M5
46		The AWT class hierarchy	**1**	T1, T2	M4
47		user interface, choices,	**1**	T1, T2	M1
48		components- labels, button, canvas, scrollbars, text components, check box, checkbox groups	**1**	T1,T2	M1,M5
49		lists panels – scrollpane, dialogs, menubar, graphics	**1**	T1, T2	M1,M5
50		layout manager – layout manager types – border, grid, flow, card and grid bag	**1**	T1, T2	M1,M5
51		**UNIT-IV**	**L46**	**SLIP TEST**	
UNIT – V					
52	**Applets**	Concepts of Applets, differences between applets and applications.	**1**	T1, T2	M1
53		life cycle of an applet	**1**	T1,T2	M4
54		types of applets,	**1**	T1, T2	M4
55		creating applets, passing parameters to applets	**1**	T1, T2	M1,M5
56		Swing – Introduction, limitations of AWT	**1**	T1, T2	M1
57		MVC architecture, components, containers	**1**	T1,T2	M4
58		exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields	**1**	T1, T2	M1,M5
59		buttons – The JButton class, Check boxes,	**1**	T1, T2	M1,M5

		Radio buttons,			
60		Combo boxes, Tabbed Panes, Scroll Panes	**1**	T1, T2	M1,M5
61		Trees, and Tables	**1**	T1, T2	M1,M5
62		**UNIT-V**	**L56**	**SLIP TEST**	
Total			**56**		

METHODS OF TEACHING:

M1 : Lecture Method	**M4 : Presentation /PPT**	**M7 : Assignment**
M2 : DemoMethod	**M5 : Lab/Practical**	**M8 : Industry Visit**
M3 : Guest Lecture	**M6 : Tutorial**	**M9 : Project Based**

NOTE:

1. AnySubjectinaSemesterissupposetobecompletedin55to65periods.
2. Each Period is of 50minutes.
3. Each unit duration &completion should be mentioned in the Remarks Column.
4. ListofSuggestedbookscanbemarkedwithCodeslikeT1,T2,R1,R2, etc.

7. Session Execution Log:

S. no	**Unit**	**Scheduled completed date**	**Completed date**	**Remarks**
1	**I**	29-07-2024	24-08-2024	COMPLETED
2	**II**	29-08-2024	14-09-2024	COMPLETED
3	**III**	18-09-2024	04-11-2024	COMPLETED
4	**IV**	06-11-2024	16-11-2024	COMPLETED
5	**V**	18-11-2024	29-11-2024	COMPLETED

8. Lecture Notes – (hand written)

9.ASSIGNMENT QUESTIONS ALONG WITH SAMPLE ASSIGNMENT SCRIPTS

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

MID 1 ASSIGNMENT

ACADEMIC YEAR 2024-25

SUBJECT NAME: OOPs through JAVA

1a)Explain about various data types in java--CO-1

1b) Explain usage of "this" keyword in java--CO-1

2a)List and explain about java buzzword-CO-1

2b) what is recursion? Write a java program for finding factorial of given number using recursion- CO-1

3a)Explain various forms of inheritance with suitable examples-CO-2

3b) what is the use of "final" keyword in java -CO-2

4a)what is polymorphism? With suitable examples explain various types of polymorphism. CO-2

4b) Demonstrate creation and importing of user defined package with suitable program.-CO2

5a)Do java support multiple inheritance? If yes, explain with suitable example.-CO2

5b)what is an exception? With suitable program explain about exception handling machanism in java. - CO-3

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

MID II ASSIGNMENT

ACADEMIC YEAR 2024-25

SUBJECT NAME: OOPs through JAVA

1. A) Write Short note on thread life cycle? .[CO3]
   B) Explain how many ways a thread can be created with suitable example? [CO3]

2. A) Describe producer-consumer pattern using inter-thread communication[CO3]
   B) Does Java support thread priorities? Justify your answer with suitable example.[CO3]

3. A)Explain any Three Layout managers with program and output [CO4]
   B) Write about Delegation Event model. [CO4]

4. A) Write a java program to demonstrate the handling mouse events? [CO4]
   B) What is the advantage of adaptor classes? Write about any Adaptor class with example program [CO4]

5. A)Explain differences between i)AWT and Swing components ii) Application and Applet
   B) Briefly explain about the following: a) JRadio Button b) JCheckBox [CO5]

10.Mid exam question papers along with sample answer scripts

MID-1

CMR ENGINEERING COLLEGE
EXPLORE TO INVENT

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

NATIONAL BOARD OF ACCREDITATION

II-B.TECH-I SEM(R22) I -MIDEXAMINATIONS,SEP/OCT-2024
SUBJECT:OBJECTORIENTEDPROGRAMMINGTHROUGHJAVA(R22CS304PC)

SET-I

Branch:CSE,AI&ML,DS,CS&IT Marks:30 M
Date:04/10/2024
Time:10:00AMto11:30A
M

Note:Questionpapercontainstwoparts,Part-AandPart-B.
Answer all questions in
Part-A. Answeranyfour
questionsinPart-B

PART-A 5x2M=10M

		BTL	CO
1.	WhatisaClassandObject?	1	1
2.	DistinguishbetweenMethodOverloadingandMethodOverriding?	4	2
3.	ExplainaboutExceptionitstypes	2	3
4.	OutlineallJavaBuzzwords?	2	1

5.	DiscussaboutAbstractclassesinjava?	6	2

PART-B 4x5M=20M

		BTL	**CO**
6.	a)DemonstratehowareArraysDeclaredandInitializedinJavawithsuitableprogram?	2	1
	b)ListthevariousprimitiveDatatypesavailableinjavaExplain?	1	1
7.	a) Whatisrecursion?Constructajava programtofindthefactorialofgivennumber using recursion?	3	1
	b) WhatisTypecastingandexplainitstypeswithexampleprogram.	1	1
8.	DiscusaboutconstructorandconstructoroverloadingWithExampleprograminJava?	6	2
9.	Definepackage?Howtocreateaccessandimportthepackagein Java?Explainwith an Example?	1	2
10	WhatisInheritance?DiscussvariousformsofInheritancewithSuitableexample?	1	2
11	ListvariouskeywordsusedtohandleException andwriteaprogramtohandleanexception?	4	3

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

CMR
ENGINEERING COLLEGE
EXPLORE TO INVENT

NATIONAL BOARD OF ACCREDITATION

NAAC

II-B.TECH-ISEM(R22)I-MIDEXAMINATIONS,SEP/OCT-2024

SUBJECT: OBJECT ORIENTED PROGRAMMINGTHROUGH JAVA(R22CS304PC)

SET-II

Branch:CSE,AI&ML,DS,CS&IT Marks:30 M
Date:04/10/2024
Time:10:00AMto11:30AM--

Note:Questionpapercontainstwoparts,Part-AandPart-B. Answer all questions in Part-A. AnsweranyfourquestionsinPart-B

PART-A 5x2M=10M

		BTL	CO
1.	ListalldifferentDatatypeusedinJava?	1	1
2.	Writeabout“Super”keyword?	1	2
3.	Listoutvariouskeywordsusedtohandleexceptionsin java	1	3
4.	Distinguishbetweenabstractclassandinterface?	4	2
5.	ListStringManipulationfunctionsofJavaStringclass?	4	1

PART-B 4x5M=20M

		BTL	CO
6.	a)ExplainbrieflyabouttheBuzzwordsofJava?	5	1
	b)Explainabout“for each”loopingstatementinJavawith Examples?	5	1
7	Define polymorphism? Compare and contrast Method overloadingandMethodOverridingwithanexampleprogram?	5	2
8	WhatisInterface?Writea programtodemonstratehowdo interface is used to achieve multiple inheritance in java?	1	2
9	DiscusvariousformsofinheritanceinJavawithExamples?	6	2
10	Whatisa Stream?Writea javaprogramtoreadinputthrough console class.	1	2
11	Whataredifferenttypes ofException andconstructa javaprogramto demonstrateanExceptionhandlingmechanismwithsuitableexample?	3	3

CMR ENGINEERING COLLEGE
EXPLORE TO INVENT
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

II-B.TECH-I SEM(R22) I -MIDEXAMINATIONS,SEP/OCT-2024

SUBJECT:OBJECTORIENTEDPROGRAMMINGTHROUGHJAVA(R22CS304PC)

SET-III

Branch: CSE, AI&ML, DS, CS &IT Marks:30M Date:04/10/2024

Note:Questionpapercontainstwoparts,Part-AandPart-B. Answer all questions in Part-A. AnsweranyfourquestionsinPart-B

PART-A 5x2M=10M

		BTL	CO
1.	Whyjava isplatformindependent?	1	1
2.	ExplainOOPsconcepts.	1	2
3.	Whatisarecursion?	1	2
4.	DistinguishbetweenExceptionandError?	4	1
5.	Whatistheneedofgarbagecollectorinjava?	1	1

PART-B 4 x5M=20M

		BTL	CO
6.	a)ExplainbrieflyabouttheBuzzwordsofJava?	5	1
	b)Definepolymorphism?Explaindifferenttypesof polymorphism with an example program?	1	1
7	Whatispackage?Howtocreateuserdefinedpackageandimport ?Explainwithanexampleprogram?	1	1
8	a)HowareArraysDeclaredandInitializedin Java?Explainwithsuitable Java Program?	1	2
	b)ExplainaboutvariousDatatypesavailableinjava?	1	1
9	DiscusvariousformsofinheritanceinJavawithExamples?	6	2
10	WriteaboutBytestreamandcharacterstream?Writeajava program to read input through console class.	1	2
11	WhatisException?Writea programtodemonstrateexceptionhandling mechanism in java.	1	3

CMR
ENGINEERING COLLEGE
EXPLORE TO INVENT

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

NBA
NATIONAL BOARD of ACCREDITATION

NAAC

II-B.TECH-I SEM(R22) II -MIDEXAMINATIONS,DEC-2024

SUBJECT:OBJECTORIENTEDPROGRAMMINGTHROUGHJAVA(R22CS304PC)

SET-I

Branch: CSE, AI&ML, DS, CS &IT Marks:30M

Date:06/12/2024 Time: 10:00AM to 11:30AM

Note:Questionpapercontainstwoparts, Part-AandPart-B.

Answer all questions In Part-A.

Answeranyfourquestionsin Part-B

1.	WhatareadvantagesofusingGenerics?	1	3
2.	DiscussdifferencebetweenMultithreadingandMultitasking	6	3
3.	ListlimitationofusingAWT components	1	4
4.	ComparePanelandFrame.	4	5
5.	IllustrateLifecycleofApplet.	2	5

PART-B 4x5M=20M

		BTL	**CO**
6.	ExplainaboutInterThreadCommunicationwithansuitable example.	2	3
7	List out how to many ways a mutual exclusion can be achieved using Threads. Explain any one with suitable example.	1	3
8	a) DevelopajavaprogramtodemonstratethehandlingKeyEvents . b) DevelopajavaprogramonKeyAdapterclasseswithexample?	3	4
9	List out various Layout Managers available in java and Explain any two layout managers with program.	4	4

10	ExplainanyThreeswingcomponents?	5	5
11	Explainpassing parameterstoapplets withanExample.	5	5

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

II-B.TECH-I SEM(R22) II -MIDEXAMINATIONS,DEC-2024

SUBJECT:OBJECTORIENTEDPROGRAMMINGTHROUGHJAVA(R22CS304PC)

SET-II

Branch: CSE, AI&ML, DS, CS &IT Marks:30M Date:06/12/2024

Note:Questionpapercontainstwoparts, Part-AandPart-B.

Answer all questions In Part-A.

Answeranyfourquestionsin Part-B

1.	Whatisautoboxingandunboxinginjava	1	3
2.	WhatisthepurposeofThread group	1	3
3.	DefinethetermsEvent,Sourceand Listeners	1	4
4.	CompareandcontrastApplicationand Applet	4	5
5.	BrieflyexplainaboutJRadioButton	5	5

PART-B 4x5M=20M

		BTL	**CO**
6.	Illustrate about a) thread lifecycleb)Appletlifecycle	2	3&5
7	Constructa program tosolveproducer-consumerproblem by using Thread concept	6	3
8	a) DemonstrateMouseEventswithanexample b) Constructajavaprogramonmouseadapterclasseswith example?	2&3	4
9	DiscussanyTwoLayoutmanagerswithprogramandoutput	6	5

10	ListanyThreeswingcomponentsandexplainwithsuitable program?	1	5
11	ExplainwithaneatdiagramaboutMVC architecture	5	5

CMR ENGINEERING COLLEGE
EXPLORE TO INVENT
CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

II-B.TECH-I SEM(R22) II -MIDEXAMINATIONS,DEC-2024

SUBJECT:OBJECTORIENTEDPROGRAMMINGTHROUGHJAVA(R22CS304PC)

SET-III

Branch: CSE, AI&ML, DS, CS &IT Marks:30M Date:06/12/2024

Note:Questionpapercontainstwoparts, Part-AandPart-B.

Answer all questions In Part-A.

AnsweranyfourquestionsinPart-B

1.	IllustrateLifecycleofThread.	2	3
2.	WhatisthepurposeofEnumerationinjava	1	3
3.	Defineinnerclassesand nestedinnerclass.	1	4
4.	ExplainaboutDelegationEventmodel	5	4
5.	CompareandcontrastAWTandSwing injava?	4	5

PART-B 4x5M=20M

		BTL	**CO**
6.	Explainthevariouswaysofcreatingthreadwithsuitable programs	2	3
7	Discusshowthread canbesynchronized withexample.	6	3
8	DoesJavasupportthreadpriorities?Justifyyouranswerwith suitable example	5	3
9	Demonstrate aboutFlowandBorderLayoutmanagerswith program	2	4
10	Brieflyexplainaboutthefollowing:a)JRadioButtonb) JCheckBox	2	5

11	Discuss aboutCreatingofSwingAppletwithanexampleprogram.	6	5

11. SCHEME OF EVALUATION

Mid-1 SET-I

S.NO	THEORY	MARKS	TOTAL
1	**Definition**	2	10
2	**Distinguish**	2	
3	**Explain**	2	
4	**Outline**	2	
5	**Discuss**	2	
6	**Part-B** **a. Demonstrate**	3	20
	b. **List**	2	
7	**a. Define** and **Construct**	3	
	b. **Define**	2	
8	**Discuss**	5	
9	**Define** and **Construct**	5	
10	**Definition** and **Explaination?**	5	
11	**List**	5	
TOTAL MARKS		**40**	**30**

SCHEME OF EVALUATION

MID-1 SET-II

S.NO	THEORY	MARKS	TOTAL
1	**List**	2	10
2	**Write**	2	
3	**List**	2	
4	**Distinguish**	2	
5	**List**	2	
6	a)**Explain**	3	20
	b)**Explain**	2	
7	**Define**	5	
8	**Define** and **Construct**	5	
9	**Discuss**	5	
10	**Define**	5	
11	**Explain**	5	
TOTAL MARKS		**40**	30

SCHEME OF EVALUATION

MID-1 SET-III

S.NO	THEORY	MARKS	TOTAL
1	**Justify**	2	10
2	**Explain**	2	
3	**Definition**	2	
4	**Distinguish**	2	
5	**Definition**	2	
6	**a)Explain**	3	20
	b)Define and **Explain**	2	
7	**Define**	5	
8	a)**Explain**	3	
	b)**Explain**	2	
9	**Discuss**	5	
10	**Write**	5	
11	**What** and **demonstrate**	5	
TOTAL MARKS		40	**30**

MID-2 SET-I

SCHEME OF EVALUATION

S.NO	THEORY	MARKS	TOTAL
	Definition-	2	10

1			
2	**Definition and Purpose**	2	
3	**Definition**	2	
4	**Compare**	2	
5	**Explanation**	2	
6	**Illustrate**	3	20
	illustrate	2	
7	**Program**	5	
8	**Demonstrate**	3	
	Construct	2	
9	**Discuss**	5	
10	**List**	5	
11	**Explain**	5	
TOTAL		**40**	**30**

SCHEME OF EVALUATION

S.NO	THEORY	MARKS	TOTAL

1	**Part-A** **Working**	2	10
2	**Define**	2	
3	**Write**.	2	
4	**List**.	2	
5	**Define** and **List**	2	
6	**Part-B** **Discuss**	5	20
7	**Demonstrate**	5	
8	**Construct**	5	
	Discuss		
9	**List**	5	
10	**Explain**	5	
11	**Explain**	5	
TOTAL MARKS		**40**	30

SCHEME OF EVALUATION

S.NO	THEORY	MARKS	TOTAL
1	**Illustrate**	2	10

2	**Definition**	2	
3	**Definition**	2	
4	**Explain**	2	
5	**Compare**	2	
6	**Explain**	5	20
7	**Discuss**	5	
8	**Justify**	5	
9	**Demonstrate**	5	
10	**Explain**	5	
11	**Discuss**	5	
TOTAL MARKS		**40**	**30**

12.Mapping of COs and Pos with PSOs*********PO 7 HAS REMOVED BY NBA

KINDLY DON'T MAP

COURSE													
CO-PO&PSO MATRIX	**PO 1**	**PO 2**	**P O3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PS O1**	**PS O2**
CO1	2		2	2									
CO2		2		2								2	
CO3			3									2	
CO4	3		3	2								2	
CO5	3				2							2	1
AVERAGE	**3**	**2**	**3**	**2**	**2**							**2**	**1**

Mapping of POs with PEOs

	Program Outcome(PO):											
PEOS		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
	I	x	x		x							
	II		x	x			x					
	III							x				x
	IV						x			x		

13.COs,POs,PSOs JUSTIFICATION

COURSE OUTCOMES

CO1	Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection **[BTL1,BTL2,BTL3][REMEMBERING,UNDERSTANDING,APPLYING]**
CO2	Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implements keywords. **[BTL2,BTL3][UNDERSTANDING,APPLYING]**
CO3	Use multithreading concepts to develop inter process communication. **[BTL2,BTL6][UNDERSTANDING,CREATING]**
CO4	Understand the process of graphical user interface design and implementation using AWT or swings..**[BTL2,BTL3,BTL6][UNDERSTANDING,APPLYING,CREATING]**
CO5	Develop applets that interact abundantly with the client environment and deploy on the server..**[BTL3,BTL6][APPLYING,CREATING]**

Justification:

COURSE CO-PO&PSO-MATRIX	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	-	2	2	-	-	-	-	-	-	-	-	-

CO1 :Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection
Correlated with PO-1 is moderate: CO1 WITH PO1: Apply the knowledge of OOPs concepts such as inheritance polymorphism, abstraction and encapsulation for solving complex engineering problems.
Correlated with PO3 is moderate: Identify and analyze usage OOPs concepts to solve the problems.
Correlated with PO4 is moderate: By the knowledge on OOPs student will be able toanalysisand interpretthe data and able to provide the valid conclusions about the problemstatement.

COURSE CO-PO&PSO-MATRIX	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO2	-	2	-	2	-	-	-	-	-	-	2	-	-

CO2: Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implements keywords

Correlated with PO2: Identify and analyze usage of interface, packages and handling of files concepts to solve the problems.

Correlated with PO4 is moderate:with the knowledge on interface, packages and handling of files student will be able to design, analyze and interpret data which will be helpful in investigation ofcomplexproblem.

Correlated with PSO1 is moderate.: This provides the knowledge and also helps to develop a solution for real-world problems by using Stream based IO,packages and interfaces .

COURSE CO-PO&PSO-MATRIX	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO3			3									2	

CO3: Use multithreading concepts to develop inter process communication

Correlated withPO3is high.: It contributes only knowledge on thread based multitasking and exception handling which will be helpful in design and develop the solutions for multitasking applications with properhandling of unexpected exceptions in applications.

Correlated with PSO1 is moderate.: This provides the knowledge and also helps to develop a solution for real-world problems by using Exception handling and multithreading concepts .

COURSE CO-PO&PSO-MATRIX	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO4	3		3	2								2	

CO 4: Understand the process of graphical user interface design and implementation using AWT or swings..**[**

Correlated with PO1is high: Apply the knowledge of GUI programming ,Event handling, applets and swings in solving complex engineering problems.

Correlated with PO3 is high: Student uses the tools such as Layout Managers, Swings controls and applets in predicting and modeling the complex activities with knowledge on limitations of AWT.

Correlated with PSO1 is moderate.: Student is able to analyze design and develop the WEB based applications with good lookand feelswingcomponents and controls.

COURSE CO-PO&PSO-MATRIX	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO5	3				2							2	1

CO 5: Develop applets that interact abundantly with the client environment and deploy on the

server
Correlated with PO1is high: Apply the knowledge of GUI programming ,Event handling, applets and swings in solving complex engineering problems.
Correlated with PO5 is moderate: Student uses the tools such as Layout Managers, Swings controls and applets in predicting and modeling the complex activities with knowledge on limitations of AWT.
Correlated with PSO1 is moderate.: Student is able to analyze design and develop the WEB based applications with good lookand feelswingcomponents and controls.
Correlated with PSO2 is low.: Student use knowledge of GUI, Event handling, Applet and Swings in rendering solution with innovative ideas.

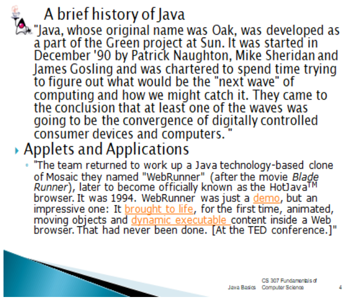
<u>14.Attainment of COs,POs AND PSOs (Excel sheet)</u> AFTER RESULT

15. Previous Question Papers

<u>16.Power point presentations (PPTs)</u>

Brief History of Java and Overview of Langauge

java.sun.com/features/1998/05/birthday.html

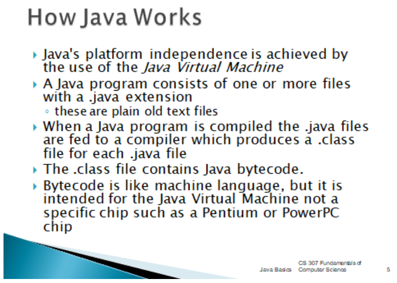


A brief history of Java

"Java, whose original name was Oak, was developed as a part of the Green project at Sun. It was started in December '90 by Patrick Naughton, Mike Sheridan and James Gosling and was chartered to spend time trying to figure out what would be the "next wave" of computing and how we might catch it. They came to the conclusion that at least one of the waves was going to be the convergence of digitally controlled consumer devices and computers. "

- Applets and Applications
  - "The team returned to work up a Java technology-based clone of Mosaic they named "WebRunner" (after the movie *Blade Runner*), later to become officially known as the HotJava™ browser. It was 1994. WebRunner was just a demo, but an impressive one: It brought to life, for the first time, animated, moving objects and dynamic executable content inside a Web browser. That had never been done. [At the TED conference.]"

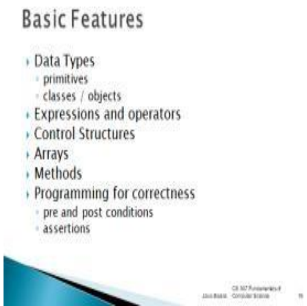
Java Basics CS 307 Fundamentals of Computer Science 4



How Java Works

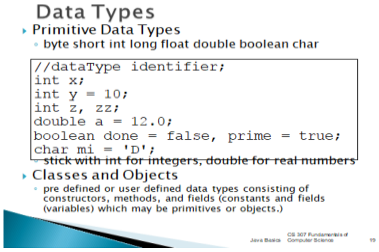
- Java's platform independence is achieved by the use of the *Java Virtual Machine*
- A Java program consists of one or more files with a .java extension
  - these are plain old text files
- When a Java program is compiled the .java files are fed to a compiler which produces a .class file for each .java file
- The .class file contains Java bytecode.
- Bytecode is like machine language, but it is intended for the Java Virtual Machine not a specific chip such as a Pentium or PowerPC chip

Java Basics CS 307 Fundamentals of Computer Science 5



Basic Features

- Data Types
  - primitives
  - classes / objects
- Expressions and operators
- Control Structures
- Arrays
- Methods
- Programming for correctness
  - pre and post conditions
  - assertions



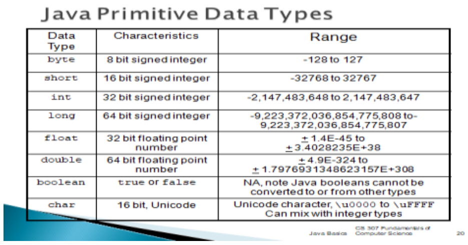
Data Types

- Primitive Data Types
  - byte short int long float double boolean char

```
//dataType identifier;
int x;
int y = 10;
int z, zz;
double a = 12.0;
boolean done = false, prime = true;
char mi = 'D';
```

  - ~~stick with int for integers, double for real numbers~~
- Classes and Objects
  - pre defined or user defined data types consisting of constructors, methods, and fields (constants and fields (variables) which may be primitives or objects.)

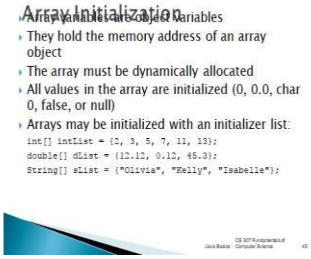
Java Basics CS 307 Fundamentals of Computer Science 19



Java Primitive Data Types

Data Type	Characteristics	Range
`byte`	8 bit signed integer	-128 to 127
`short`	16 bit signed integer	-32768 to 32767
`int`	32 bit signed integer	-2,147,483,648 to 2,147,483,647
`long`	64 bit signed integer	-9,223,372,036,854,775,808 to- 9,223,372,036,854,775,807
`float`	32 bit floating point number	± 1.4E-45 to ± 3.4028235E+38
`double`	64 bit floating point number	± 4.9E-324 to ± 1.7976931348623157E+308
`boolean`	`true` or `false`	NA, note Java booleans cannot be converted to or from other types
`char`	16 bit, Unicode	Unicode character, `\u0000` to `\uFFFF` Can mix with integer types

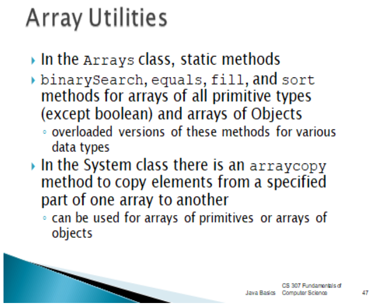
Java Basics CS 307 Fundamentals of Computer Science 20



Array Initialization

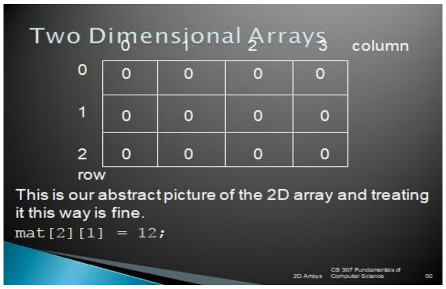
- Array variables are object variables
- They hold the memory address of an array object
- The array must be dynamically allocated
- All values in the array are initialized (0, 0.0, char 0, false, or null)
- Arrays may be initialized with an initializer list:

```
int[] intList = {2, 3, 5, 7, 11, 13};
double[] dList = {12.12, 0.12, 45.3};
String[] sList = {"Olivia", "Kelly", "Isabelle"};
```



Array Utilities

- In the `Arrays` class, static methods
- `binarySearch`, `equals`, `fill`, and `sort` methods for arrays of all primitive types (except boolean) and arrays of Objects
  - overloaded versions of these methods for various data types
- In the System class there is an `arraycopy` method to copy elements from a specified part of one array to another
  - can be used for arrays of primitives or arrays of objects

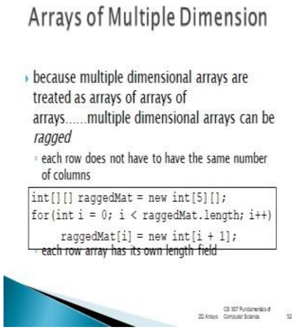


Two Dimensional Arrays

row	0	1	2	3 column
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0

This is our abstract picture of the 2D array and treating it this way is fine.

```
mat[2][1] = 12;
```

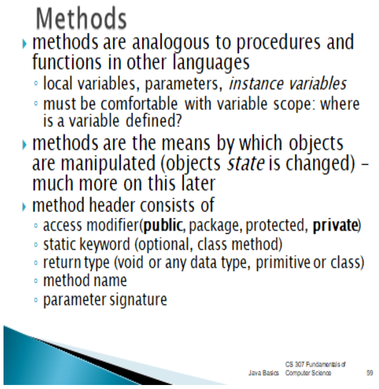


Arrays of Multiple Dimension

- because multiple dimensional arrays are treated as arrays of arrays of arrays......multiple dimensional arrays can be *ragged*
  - each row does not have to have the same number of columns

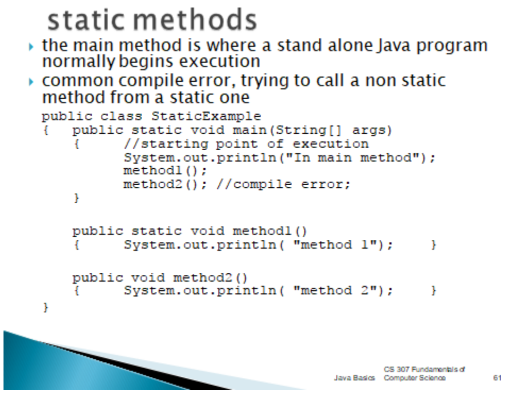
```
int[][] raggedMat = new int[5][];
for(int i = 0; i < raggedMat.length; i++)
     raggedMat[i] = new int[i + 1];
```

  - each row array has its own length field



Methods

- methods are analogous to procedures and functions in other languages
  - local variables, parameters, *instance variables*
  - must be comfortable with variable scope: where is a variable defined?
- methods are the means by which objects are manipulated (objects *state* is changed) – much more on this later
- method header consists of
  - access modifier(**public**, package, protected, **private**)
  - static keyword (optional, class method)
  - return type (void or any data type, primitive or class)
  - method name
  - parameter signature



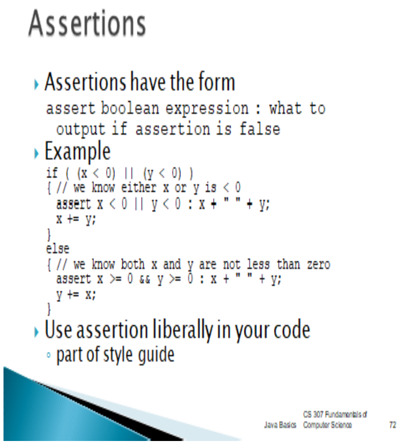
static methods

- the main method is where a stand alone Java program normally begins execution
- common compile error, trying to call a non static method from a static one

```
public class StaticExample
{   public static void main(String[] args)
    {     //starting point of execution
          System.out.println("In main method");
          method1();
          method2(); //compile error;
    }

    public static void method1()
    {     System.out.println( "method 1");    }

    public void method2()
    {     System.out.println( "method 2");    }
}
```



Assertions

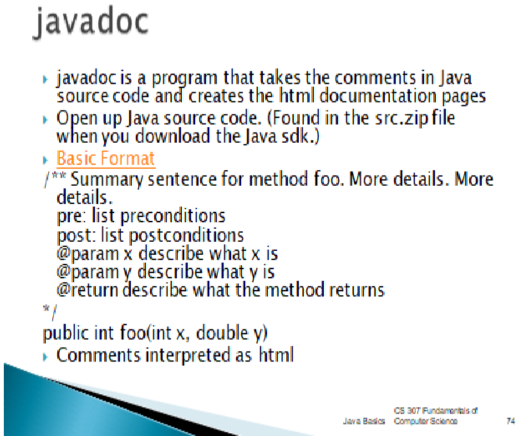
- Assertions have the form

```
assert boolean expression : what to
 output if assertion is false
```

- Example

```
if ( (x < 0) || (y < 0) )
{ // we know either x or y is < 0
  assert x < 0 || y < 0 : x + " " + y;
  x += y;
}
else
{ // we know both x and y are not less than zero
  assert x >= 0 && y >= 0 : x + " " + y;
  y += x;
}
```

- Use assertion liberally in your code
  - part of style guide



javadoc

- javadoc is a program that takes the comments in Java source code and creates the html documentation pages
- Open up Java source code. (Found in the src.zip file when you download the Java sdk.)
- Basic Format

```
/** Summary sentence for method foo. More details. More
details.
pre: list preconditions
post: list postconditions
@param x describe what x is
@param y describe what y is
@return describe what the method returns
*/
public int foo(int x, double y)
```

- Comments interpreted as html

17. Innovative Teaching method if any(Attached Innovative Assignment)

QUESTIONS

1. What is meant by byte code? Briefly explain how Java is platform independent?

2. Explain the significance of public, protected and private access specifiers in inheritance.

3. Write the significance of Java Virtual Machine.

4. How do we implement polymorphism in JAVA? Explain briefly.

5. How to design and implement an interface in java? Give an example.

18. References (Textbook/Websites/Journals)

Textbooks

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.[T1]

2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education. [T2]

Websites or URLs e- Resources

a. http://www.cmrec.ac.in/ac17-18materialCse.html

b. https://onlinecourses.nptel.ac.in/noc21_cs03/preview

c. https://www.coursera.org/specializations/object-oriented-programming

d. http://www.btechsmartclass.com/

e. http://www.javatpoint.com/

f. https://www.tutorialspoint.com/

Journals

g. An interactive environment for beginning Java programmers

h. Systems programming in Java

i. The case for Java as a programming language

j. Java in real-time applications

k. Concept–based Analysis of Java Programming Errors among Low, Average and High Achieving Novice Programmers