ISO 9001:2015 Certified Institution Estd.:2001

BITS AUTONOMOUS

Balaji Institute of Technology & Science

Laknepally (V), Narsampet (M), Warangal District - 506 331, Telangana State, India
(AUTONOMOUS)
Accredited by NBA (UG - CE, ME, ECE & CSE) & NAAC A+ Grade
(Affiliated to JNT University, Hyderabad and Approved by AICTE, New Delhi)
www.bitswgl.ac.in, email: principal@bitswgl.ac.in, Ph:98660 50044, Fax: 08718-230521

Course File Contents:

I.COVER PAGE

BALAJI INSTITUTE OF TECHNOLOGY & SCIENCE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING(AIML)

Name of the Subject: Scripting Langauages

JNTU CODE : Programme : **UG**

Branch:**CSE(AIML)**	Version No :
Year: **II**	Document Number :**BITS/CSM/**
Semester: **I**	Number of Pages :

Classification status (Unrestricted/Restricted) : Unrestricted

Distribution List: Dept. Library, Dept. Office, Concerned Faculty

Prepared by :	Updated by :
1. Name : **P.Rajesh**	**1.** Name :
2. Sign :	**2.** Sign :
3. Design: **Assistant Professor**	**3.** Design :
4. 4) Date:	**4.** Date :

Verified by :*For Q.C only

1. Name :	**1.** Name :
2. Sign :	**2.** Sign :
3. Design :	**3.** Design :
4. Date :	**4.** Date :

Approved by (HOD) :

1. Name :

2. Sign :

3. Date :

2.VISION AND MISSION OF THE DEPARTMENT

Vision: To be a global leader in Artificial Intelligence and Machine Learning research, innovation, and education, driving transformative advancements that empower industries, enhance human capabilities, and contribute to a smarter, more sustainable world.

M1:Innovative Research& Quality Education – To Conduct research on cutting-edge Technologies to address complex real-world problems across diverse domains and provide world-class education and training to equip students with technical expertise, ethical responsibility, and problem-solving skills.

M2: Industry Collaboration & Ethical AI Development –To Foster strong partnerships with industries, academia, and government organizations to develop impactful AI solutions and promote responsible and ethical AI practices that align with societal values and global sustainability.

M3: Entrepreneurship & Innovation – Encourage entrepreneurship and the development of AI-driven start-ups and products that contribute to economic growth.

M4: Community Engagement – Engage with communities to spread AI awareness, inclusivity, and accessibility for societal benefit.

3. PEOs, POs and PSOs

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The objectivesforECEgraduates are to:

- ❖ **PEO1:** Graduates shall apply the analytical, decision making and prediction skills in AI & ML to formulate and solve complex intelligent computing and multidisciplinary problems.
- ❖ **PEO2**:Graduates will be able to take up higher studies, research & development by acquiring in-depth knowledge in Artificial Intelligence & Machine Learning.
- ❖ **.PEO3**: Graduates will be able to exhibit their employability skills and practice the ethics of their profession with a sense of social responsibility..

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

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. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **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.
12. **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.

BALAJI INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS)

21CS670PE: SCRIPTING LANGUAGES (PE-II)

4. Syllabus and Academy Calendar

UNIT - I

Introduction: Ruby, Rails, The structure and Execution of Ruby Programs, Package Management with RUBYGEMS, Ruby and web: Writing CGI scripts, cookies, Choice of Webservers, SOAP and web services

RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

UNIT - II

Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

UNIT - III

Introduction to PERL and Scripting

Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL-Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT - IV

Advanced perl

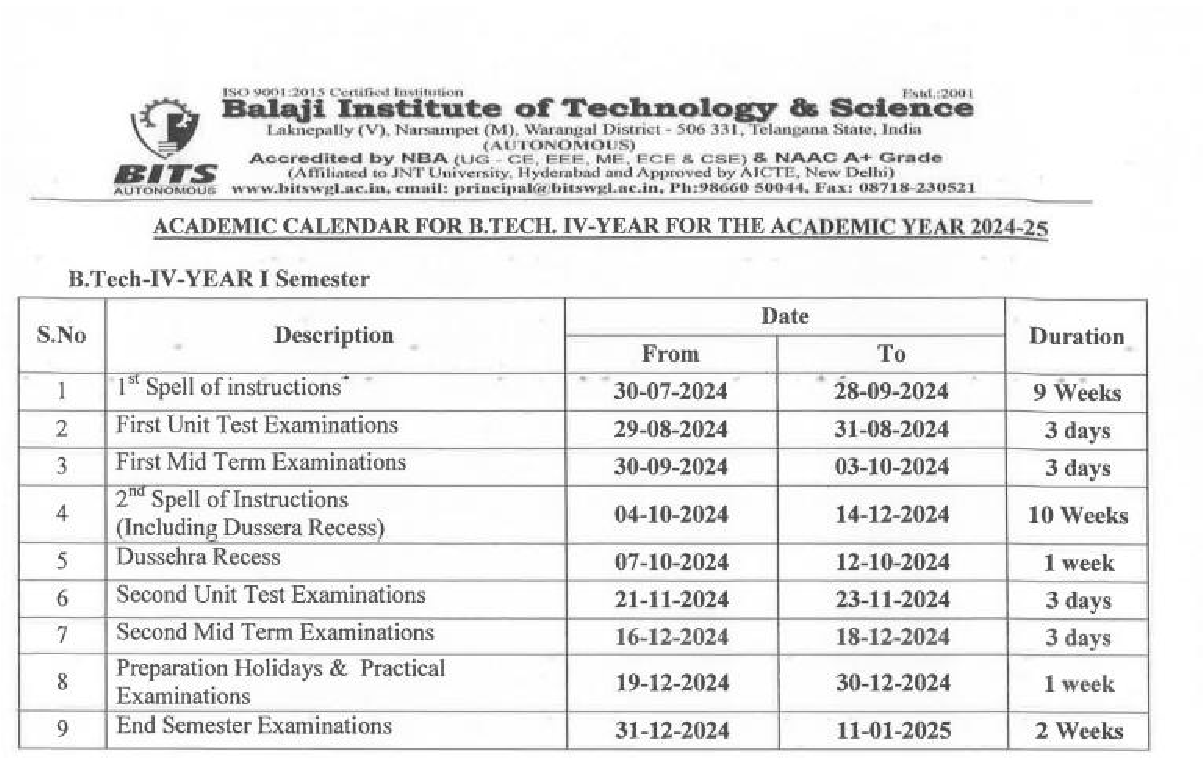
Finer points of looping, pack and unpack, filesystem, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT-V

TCL

TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security

Issues, C Interface. Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.



ISO 9001:2015 Certified Institution Estd.:2001

BITS
AUTONOMOUS

Balaji Institute of Technology & Science
Laknepally (V), Narsampet (M), Warangal District - 506 331, Telangana State, India
(AUTONOMOUS)
Accredited by NBA (UG - CE, EEE, ME, ECE & CSE) **& NAAC A+ Grade**
(Affiliated to JNT University, Hyderabad and Approved by AICTE, New Delhi)
www.bitswgl.ac.in, email: principal@bitswgl.ac.in, Ph:98660 50044, Fax: 08718-230521

ACADEMIC CALENDAR FOR B.TECH. IV-YEAR FOR THE ACADEMIC YEAR 2024-25

B.Tech-IV-YEAR I Semester

S.No	Description	Date		Duration
		From	To	
1	1st Spell of instructions	30-07-2024	28-09-2024	9 Weeks
2	First Unit Test Examinations	29-08-2024	31-08-2024	3 days
3	First Mid Term Examinations	30-09-2024	03-10-2024	3 days
4	2nd Spell of Instructions (Including Dussera Recess)	04-10-2024	14-12-2024	10 Weeks
5	Dussehra Recess	07-10-2024	12-10-2024	1 week
6	Second Unit Test Examinations	21-11-2024	23-11-2024	3 days
7	Second Mid Term Examinations	16-12-2024	18-12-2024	3 days
8	Preparation Holidays & Practical Examinations	19-12-2024	30-12-2024	1 week
9	End Semester Examinations	31-12-2024	11-01-2025	2 Weeks

5. Brief notes on the importance of the course

The importance of a **course scripting language program** can be highlighted through the following points:

1. **Hands-On Learning**: A scripting language program offers practical experience in coding, allowing students to apply theoretical concepts in real-world scenarios, enhancing their learning.
2. **Simplifies Complex Tasks**: Scripting languages are often designed to simplify complex programming tasks, making them more accessible to beginners while still being powerful enough for advanced tasks.
3. **Increases Efficiency**: Students can automate repetitive tasks, manage data more effectively, and streamline workflows, making their work more efficient and less time-consuming.
4. **Improves Problem-Solving**: Writing scripts forces students to break down problems logically, improving their problem-solving and critical-thinking abilities, essential skills in both academic and professional environments.
5. **Versatile Skillset**: Learning a scripting language helps students develop a versatile skill set that can be applied across different fields such as web development, data analysis, automation, and more.
6. **Real-Time Feedback**: Scripting programs often allow for immediate feedback (through execution), enabling students to quickly identify errors and refine their code, fostering a better understanding of programming logic.
7. **Preparation for Advanced Topics**: Scripting languages lay the groundwork for learning more complex programming languages and technologies, making them an essential stepping stone for advanced computer science topics.
8. **Industry-Relevant Skills**: Scripting languages like Python, JavaScript, and Bash are in high demand in the tech industry. A course in scripting languages equips students with skills that are highly valued in the job market.
9. **Encourages Collaboration**: Scripting languages are often used in collaborative projects, such as open-source software development or team-based projects, teaching students how to work together and contribute to shared goals.

6. Prerequisites:

1. Basic Programming Concepts:

- **Variables and Data Types:** Understanding how to store and manipulate different types of data (e.g., numbers, strings, arrays/lists).
- **Basic Input/Output:** Knowing how to interact with the user, either through reading input or displaying output (e.g., puts in Ruby, print in Perl/TCL).

2. Knowledge of Basic Data Structures

Understanding basic data structures will make learning scripting languages much easier:

- **Arrays**: Arrays are fundamental in Ruby, Perl, and TCL, though each language handles them slightly differently. In all three languages, arrays store lists of elements that can be accessed by index.
- **Strings**: Strings are essential for text manipulation in all three languages. Each language has its own way of handling strings (e.g., string interpolation in Ruby, regular expressions in Perl).
- **Hashes/Associative Arrays**: A hash (also called an associative array or dictionary) is a collection of key-value pairs. In Ruby, it’s a **Hash**, in Perl, it's a **hash**, and in TCL, it's usually handled via **arrays** where the index is a string.

3. Text Processing and Regular Expressions

- Perl is especially known for its text-processing capabilities, and Ruby and TCL also have strong support for regular expressions. Understanding regular expressions is crucial in any of these languages, particularly for tasks involving text manipulation and pattern matching.

7.Course Objectives and Course Outcomes

Course Objectives:

- This course introduces the script programming paradigm
- Introduces scripting languages such as Perl, Ruby and TCL.
- Learning TCL
- Comprehend the differences between typical scripting languages and typical system and application programming languages.
- Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select an appropriate language for solving a given problem.
- Acquire programming skills in scripting language

Course Outcomes:

Course Name: C316 (SCRIPTING LANGUAGES - 22CS682PE) **After Completion of the course student will be able to: Year of Study: 2024-2025**	
C316.1	**Understand** the fundamentals of Ruby programming, Rails framework, web scripting, and GUI development with RubyTk.
C316.2	**Develop** skills to extend Ruby using C, memory management, and embedding Ruby in other languages.
C316.3	**Gain** knowledge of Perl scripting, its characteristics, and usage in web development.
C316.4	**Implement** advanced Perl programming techniques, including data structures, OS interfacing, and security concerns.
C316.5	**Apply** TCL scripting techniques, event-driven programming, and security concepts for internet-aware applications.

8. CO-PO and CO-PSO Mapping and

Justifications CO-PO Mapping

Name of the Subject: C316 SCRIPTING LANGUAGES - 22CS682PE Year of Study: 2024-2025												
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
C316.1	3	3	3	-	2	-	-	-	-	2	-	3
C316.2	3	3	3	2	3	-	-	-	2	2	2	3
C316.3	3	3	3	3	2	-*	-	-	-	-	3	2
C316.4	3	3	3	3	2	-	-	-	-	2	2	2
C316.5	3	3	3	3	2	-	-	-	2	3	3	3
Average	**3.0**	**3.0**	**3.0**	**2.7**	**2.2**	**-**	**-**	**-**	**2.0**	**2.2**	**2.5**	**2.6**

CO-PSO Mapping

Course Name: C316 SCRIPTING LANGUAGES - 22CS682PE Year of Study: 2024-2025		
CO	**PSO1**	**PSO2**
C316.1	2	-
C316.2	3	-
C316.3	3	-
C316.4	3	3
C316.5	3	3
Average	**2.8**	**3.0**

9. Department Time table & Personal time table

Day	P1	P2	P3	P4	P5	P6	P7
MON							
TUE							
WED							
THU							
FRI							
SAT							

10. Method of teaching, Chalk and talk/ppts/NPTEL lectures/cd/innovative teaching method, etc

1.Peer Teaching

- **Description**: Encourage students to teach each other, which helps reinforce their learning and improves understanding of difficult concepts.
- **Application**:
    - Students can work in groups to present topics like regular expressions in Perl or object-oriented programming in Ruby to the class.
    - This method helps students solidify their knowledge and practice communication skills.

2.Use of Virtual Labs and Simulations

- **Description**: Virtual environments that simulate real-world programming scenarios.
- **Application**:
    - Use cloud-based IDEs or virtual labs to allow students to practice coding without the need for setting up complex environments locally.
    - Example: Provide a cloud-based IDE for students to write, test, and debug Ruby, Perl, or Tcl code.

3.Innovative Teaching Methods

- **Project-based Learning**
    - Assign projects where students must build small applications using Ruby, Perl, or Tcl.
    - Example: Create a web scraper with Perl, build a command-line tool in Ruby, or automate a system task using Tcl.
    - Projects encourage hands-on practice and deeper understanding.

4.Chalk and Talk

- **Description**: Traditional method of teaching where the instructor explains concepts on the board while writing code and elaborating on syntax, logic, and functions.
- **Application**:
    - For scripting languages, the teacher can write code snippets on the board, explaining the syntax, loops, conditionals, data structures, and libraries used in Ruby, Perl, or Tcl.
    - Use this method for foundational concepts like variables, control flow, and basic file handling.

5. PowerPoint Presentations (PPTs)

- **Description**: Using slides to visually represent the core concepts, syntax, examples, and use cases.
- **Application**:
  - Present slides with key concepts, code samples, and diagrams.
  - Break down the steps in the programming logic with bullet points, and provide examples of real-world applications of Ruby, Perl, and Tcl.
  - Include visual aids, such as flowcharts and diagrams, to explain control structures and functions.

6. NPTEL (National Programme on Technology Enhanced Learning) Lectures

- **Description**: NPTEL offers video lectures and online course content on a variety of technical subjects, including scripting languages.
- **Application**:
  - Recommend students to watch relevant NPTEL courses that cover topics like scripting basics, regular expressions in Perl, Ruby for web development, and TCL for automation.
  - Students can review these videos at their own pace and then discuss the concepts in class.
  - Teachers can guide discussions and clarify doubts post-lecture.

11. Lecture schedule (without faculty name)

Department of Artificial Intelligence& Machine Learning

LESSON PLAN & DELIVERY REPORT

Subject: Scripting Languages **Class: B.Tech. IV CSM(I Sem)**

Faculty: Mr. Perugu Rajesh **Regulation:R21**

Academic Year: 2024-25 (I Semester) **Commencement of Class Work: 30.7.24**

Topics (as per syllabus)	Sub Topics	Lect. No.	Scheduled Date	Topic Delivered Date
Introduction	• About Subject & Guidelines • Vision, Mission, CO's of subject • Text & Reference Books	L1	30.7.24	
	• Ruby, Rails • The structure and Execution of Ruby	L2	31.7.24	
	• Package Management with RUBYGEMS	L3	1.8.24	
	• Writing CGI scripts cookies,Choice of Webservers,	L4	2.8.24	
Ruby and web	• SOAP and web services	L5	3.8.24	
	• RubyTk– Simple Tk Application	L6	12.8.24	
RubyTk	• widgets, Binding events, Canvas, scrolling Arrays, Operators, Expressions	L7	16.8.24	
Extending Ruby	• Ruby Objects in C, • Concepts of Classes, Objects • Constructors	L8	16.7.24	
	• the Jukebox extension	L9	17.8.24	

	• Memory allocation	L10	22.8.24	
Extending Ruby	• Ruby Type System	L11	23.8.24	
	• Embedding Ruby to Other Languages	L12	24.8.24	

Topics (as per syllabus)	**Sub Topics**	**Lect. No.**	**Scheduled Date**	**Topic Delivered Date**
Introduction to PERL and Scripting	Embedding a Ruby • Interpreter	L13	27.8.24	
	• Scripts and Programs	L14	29.8.24	
	• Origin of Scripting, Scripting Today	L15	30.8.24	
	• Characteristics of Scripting Languages	L16	31.8.24	
	• Uses for Scripting Languages	L17	2.9.24	
	• Web Scripting	L18	3.9.24	
	• universe of Scripting Languages.	L19	5.9.24	
	• Defining an Interface, Implementing Interface	L20	5.9.24	
PERL	• Names and Values	L21	6.9.24	
	• Variables	L22 L23	9.9.24	
	Scalar • Expressions	L24	10.9.24	
	• Control Structures	L25	12.9.24	

		L26	13.9.24	
	• arrays,	L27	13.9.24	
PERL	• list	L28	17.9.24	
	• hashes	L29	19.9.24	
	• string	L30	20.9.24	
	• Pattern and regular expressions	L31	21.9.24	
	• subroutines.	L32	24.9.24	
Review of Syllabus & Planning (Mid I)	• Review of theory Questions • Review of objective Questions • Plan for Mid 1 exam • Tips to get good marks	L33	25.9.24	
Mid I Schedule: 26.9.24 To 28.9.24				
Mid I Marks Distribution	• Marks Distribution • Discussion about Paper • Counsel the students(AB/got poor marks)	L34	30.9.24	
Enumerations	• Enumerations,Auto boxing,	L35	1.10.24	
	• revision	L36	4.10.24	
Advanced perl	• Finer points of looping	L37	5.10.24	
	• pack and unpack	L38	18.10.24	
	• filesystem	L39	19.10.24	
	• eval	L40	21.10.24	
	• packages	L41	22.10.24	
	• modules	L42	24.10.24	
	• objects,	L43	28.10.24	

	• interfacing to the operating system	L45	29.10.24	
Topics (as per syllabus)	**Sub Topics**	**Lect. No.**	**Scheduled Date**	**Topic Delivered Date**
TCL	• TCL Structure, syntax	L46	30.10.24	
	• Variables and Data in TCL	L47	4.11.24	
	• Control Flow	L48	5.11.24	
	• Data Structures	L49	12.11.24	
Advance TCL	• input/output, procedures	L50	15.11.24	
	• strings, patterns, files	L51	22.11.24	
	eval, source, exec • and uplevel commands	L52	28.11.24	
	• Name spaces, trapping errors event driven programs, • making applications internet aware	L53	2.12.24	
Review of Syllabus & Planning (Mid II)	• Review of theory Questions • Review of objective Questions • Plan for Mid 1 exam • Tips to get good marks	L54	03.12.24	
Mid II Schedule: 5.12.24 To 7.12.24				
Mid II Marks Distribution	• Marks Distribution • Discussion about Paper • Counsel the students (AB/got poor marks)	L55	9.12.24	

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly

3. “Programming Ruby” The PramaticProgammers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O’Reilly, SPD.

ISO 9001:2015 Certified Institution Estd.:2001

Balaji Institute of Technology & Science

Laknepally (V), Narsampet (M), Warangal District - 506 331, Telangana State, India

(AUTONOMOUS)

Accredited by NBA (UG - CE, ME, ECE & CSE) & NAAC A+ Grade

(Affiliated to JNT University, Hyderabad and Approved by AICTE, New Delhi)

BITS AUTONOMOUS www.bitswgl.ac.in, email: principal@bitswgl.ac.in, Ph:98660 50044, Fax: 08718-230521

12. Details Notes

Scripting languages are programming languages used primarily for automating tasks, controlling applications, or enhancing the functionality of other software. They are often interpreted, meaning they don't require a compilation step to run the code, and they can interact with other software systems to perform tasks dynamically.

Here's a detailed overview of scripting languages:

1. Definition and Characteristics of Scripting Languages

- **Interpreted Languages**: Scripting languages are usually interpreted rather than compiled. This means that the code is executed directly by an interpreter without needing a separate compilation process.
- **High-level**: They are often high-level, meaning they abstract away most of the low-level system operations. This makes scripting languages more user-friendly and easier to write.
- **Dynamic**: Many scripting languages are dynamically typed, meaning you don’t need to specify variable types, and the types are determined at runtime.
- **Cross-platform**: Most scripting languages are platform-independent, meaning the same script can run on different operating systems (as long as the interpreter is installed).
- **Integration with Other Software**: They can be used to interact with and control other software systems, making them particularly useful for automation and system administration.

2. Common Scripting Languages

- **Python**: One of the most popular and versatile scripting languages. Python is known for its simplicity and readability, making it an excellent choice for beginners. It's used in web development, data analysis, automation, and AI development.
- **JavaScript**: A core scripting language for web development. JavaScript is primarily used for client-side scripting in web browsers but can also be used on the server-side (via Node.js). It's integral to making web pages interactive and dynamic.

- **Ruby**: Known for its elegant syntax, Ruby is often used for web development (especially with the Ruby on Rails framework). It emphasizes simplicity and productivity.
- **Perl**: Often used for text processing, system administration, and web development. Perl has a rich set of built-in functions, making it very powerful for tasks like regular expressions and string manipulation.
- **PHP**: Primarily used for server-side web development, PHP is commonly embedded in HTML to manage dynamic content. It's often used in conjunction with databases like MySQL for creating dynamic web pages.
- **Shell Scripting (Bash, Zsh)**: Shell scripting languages like Bash are used for automating tasks in Unix/Linux systems. These scripts interact directly with the operating system to perform tasks like file manipulation, process control, and networking.
- **Lua**: A lightweight scripting language often embedded in other applications (like video games) to add extensibility and customization options without changing the core functionality.
- **Tcl**: Another embeddable language used for rapid prototyping, particularly in systems that require high-level control, such as network equipment and embedded systems.

3. Uses of Scripting Languages

- **Automation**: Scripting languages are widely used to automate repetitive tasks such as file management, data entry, and system monitoring.
- **Web Development**: Many scripting languages like JavaScript, PHP, and Python are used to create dynamic, interactive websites and web applications.
- **System Administration**: Admins use shell scripting languages like Bash to manage systems, automate backups, and perform maintenance tasks.
- **Data Analysis**: Scripting languages such as Python and R are widely used for processing, analyzing, and visualizing large datasets.
- **Game Development**: Many game engines use scripting languages like Lua to enable game developers to write flexible game logic and interactions.
- **Testing and Debugging**: Scripting is also commonly used in creating automated tests for software, allowing developers to run repetitive testing procedures to catch bugs and verify features.

4. Advantages of Scripting Languages

- **Ease of Use**: Scripting languages tend to have simpler syntax and fewer boilerplate requirements than compiled languages, which makes them easy to learn and use.
- **Rapid Development**: Due to their high-level nature and simplicity, scripting languages enable faster development and prototyping of software applications.
- **Portability**: Most scripting languages are portable, meaning they can run on different platforms (Windows, Linux, MacOS) without modification, provided that the interpreter is available.
- **Integration**: Scripting languages are great for integrating multiple software systems, automating workflows, or controlling external hardware, as they usually provide easy-to-use APIs for interacting with other systems.
- **Extensibility**: They can be embedded in other applications, which makes them valuable for enhancing or customizing third-party software.

5. Disadvantages of Scripting Languages

- **Performance**: Scripting languages are generally slower than compiled languages because they are interpreted at runtime. This can be a problem for applications that require intensive computational performance.
- **Less Control**: They typically provide less control over system resources compared to lower-level languages like C or C++.
- **Debugging**: Some dynamic languages make debugging more challenging due to their flexible nature (e.g., dynamic typing).

7. Scripting vs. Programming Languages

- **Scripting Languages**: Typically, scripting languages are used to automate tasks, control programs, or manipulate data. They often don't require a separate compilation process, making them suitable for smaller, less performance-critical tasks.
- **Programming Languages**: A general-purpose language (like C, C++, or Java) is typically compiled into machine code and used to create larger applications, systems, or software where performance is crucial.

8. Key Concepts in Scripting

- **Variables**: Most scripting languages allow you to store values in variables. These values can be strings, numbers, or even more complex data structures.
- **Functions/Procedures**: Functions allow you to encapsulate logic and reuse it. Most scripting languages support defining functions.
- **Control Structures**: These include conditional statements (if/else), loops (for, while), and other decision-making structures that help control the flow of execution.
- **Data Structures**: Scripting languages offer built-in data structures such as arrays, lists, dictionaries, and sets to organize and manipulate data efficiently.

13. Additional topics

- Covered the key Ruby concepts, including **variables and data types**, **string manipulation**, **arrays and hashes**, **basic OOP concepts**, operators and **control flow**.
- Ruby provides access control methods to restrict access to certain methods within a class. These include **public**, **private**, and **protected**.
- Ruby makes it easy to work with files. You can read from and write to files using methods like File.open, read, and write.
- **Packages** in Perl allow you to group related subroutines and variables together. A **module** is a package that is stored in a separate file.
- Perl allows you to accept **command-line arguments** using the special array @ARGV. This is helpful for creating scripts that can be run with different inputs.
- Advanced Data Structures in Scipting Languages

14. Mid exam question Papers- Theory and quiz Set-1

S. No	Question	Blooms Level
1	Describe the structure of a SOAP message and its key components.	Understand
2	Can you list some common TK-Widgets used in GUI applications?	Remembering
3	What is the structure of a Ruby object in C?	Remembering
4	Provide an example of how the Jukebox extension can be applied in a Ruby program to enhance memory management.	Applying
5	How would you evaluate the performance of a modern scripting language compared to an older one?	Applying
6	Describe the differences between popular web servers like Apache, IIS, lighttpd, etc	Understand

Set-2

S. No	Question	Blooms Level
1	Describe the role of each component (Model, View, Controller) in the MVC pattern within a Ruby on Rails application.	Understand
2	What is RubyGems, and what purpose does it serve in Ruby development?	Remembering
3	Describe how memory is allocated for variables in Ruby compared to other programming languages.	Understand
4	Given a scenario, how would you choose and implement a framework from the Jukebox extension for a specific task in a Ruby application?	Applying
5	What are some common use cases for scripts in software development?	Remembering
6	Explain how Ruby’s typing system works.	Understand

Set-3

S. No	Question	Blooms Level
1	Write short notes on ruby cookies and Sessions?	Remember
2	Describe CGI Script and what are the step involved to develop CGI Programs?	Understand
3	How can you embed a Ruby interpreter? And explain any four languages.	Apply
4	Explain the logic behind a Jukebox program and how it plays songs or media files.	Understand
5	Explain the primary differences between a script and a program in terms of execution and usage.	Understand
6	What are the main characteristics of a scripting language?	Remember

Fill in the Blanks:

Unit-1

1. Ruby supports another style of multiline comment known as __________
2. SOAP stands for _______________________________________
3. _______________________ command is used to install tk in ruby.
4. irb stands for _____________________.
5. The Ruby on Rails framework follows the __________ architecture pattern
6. The __________ widget in RubyTk is used for displaying text in a graphical user interface.
7. A______________ is typically a sequence of commands or instructions written in a scripting language.
8. 3.times { print "Ruby! " }________________________
9. The acronym CGI stands for __________ Gateway Interface, which is a standard for web servers to execute scripts.
10. In Ruby, a block of code can be defined using the __________ keyword

UNIT-2

1. The structures for the basic built-in classes are defined in ______

2. .________________________ method is to initialize the object in memory.

3. Ruby is a _________________________language.

4. Ruby Gems is a ____________________________

5. In Ruby, the range operator that includes both the starting and ending values is denoted by_______________

6. To prevent users from viewing certain files is to deny access to certain extensions by configuring the _______________ Files directive.

7. Duck typing means _______________________________________

8. The default exception constructor is called with is_____________ parameters.

9. The __________ function creates a child process as a clone of the current process.

10. A _________________ is a unit of code with its own namespace.

UNIT-3

1.____________ is the process of constructing a graphical interface from a collection ofvisual objects.

2.A ________ loop can have an explicit continue block which is executed at the end of eachnormal iteration before control returns to re-test the condition.

3.The _______________ operator creates a reference to a named variable or subroutine.

4.______________________ Sorts items by their ASCII value.

5.Inheritance is realized by including a special array _______ in the package that defines the derived class.

Multiple Choice Questions

UNIT-1

1. Which library file is used to create_makefile("my_test") [ ]

A. mkmf B. mfmk C. my_test, h D.None

2. The ________________macro will return a constant representing the C type of the given object [ ]

A. T_OBJECT B. T_STRING C. TYPE (obj) D.None

3. 3. In which activity of traditional scripting include shell scripts [ ]

A. Remote Control B .Experimental' programming C. System administration D. None

4. 4. Which of the commands recognize by Geometry manager? [ ]

A. Pack B. Grid C. Place D. All of the above

5. The Ruby interpreter parses a program as a sequence of________[ ]

A. tokens B. objects C. characters D. All

6.Which of the following is supported by Ruby?[ ]
a) Multiple Programming Paradigms b) Dynamic Type System

c) Automatic Memory Management d) All of the Mentioned

7.Which of the following languages syntax matches with the Ruby's syntax?[ ]
a) Perl b) PHP c) Java d)Tcl
8.What is the extension used for saving the ruby file?[ ]
a) .ruby extension b) .rb extension c) .rrb extensiond) None of the mentioned

9.Which of the following datatypes are valid in Ruby?[ ]
a) Numbers b) Boolean c) String d) All of the mentioned

10.Which of the following is a valid assignment operator?[ ]
a) += b) -= c) *= d) All of the mentioned

UNIT-2

1.___________________is not an immediate object in ruby [ ]

A. Fixnum B. Symbol C. true D. Pointers

2.__________________ object is used to convert integer to number [ ]

A. NUM2INT B.INT2NUM C.NUM2UINT D.INT2FIX

3. In ruby all the Global variables are prefixed with_______________________[ ]

A. @ B.! C. $ D. &

4. Scripting languages, are address which type of problems [ ]

A. off the shelf B. speed of development

C. un-time efficiency D. All of the above

5. In ruby Module, class, and method definitions, and most of Ruby's statements, include _____ structure[ ]

A. Nested B. Block C. Module D. Object

6.What does the .upcase method do?[ ]

A. Convert the string to lowercase B. Convert the string to uppercase

C. Convert only lowercase string to uppercase and vice-versa D. None of above

7. Which is not a Reserved Words in Ruby?

A.Retry B.pass **C.begin** D.ensure

8.What will be output of the given code?[ ]

my_array = [1, 2, 3, 4] print my_array

A. [1, 2, 3, 4]. B. 1234 C. Error D. None of the mentioned

10.What will be the output of the following?[ ]

array = [100, 200, 300, 400, 500]

print array[5]

A. 400 B. 500 C. Nil D. None of the mentioned

UNIT-3

1. What is the use of .capitalize method?[ ]
   A. It capitalizes the entire string B. It capitalize on the first letter of the string
   C. It capitalize the strings which are in small case D. All of the mentioned

2.Perl Programming language is ___?[ ]

A. General purpose programming language B. Dynamic programming language

C. High-level interpreted programming language D. All of these

3.The creator of Perl programming language is ___.[ ]

A. James Gosling B. Brendan Eich C.Larry Wall D.Bjarne Stroustrup

4.What is the file extension for the Perl program?[ ]

A. .pl B. .perl C. .prl D.None of these

5. What is the correct syntax for defining a class in Perl?[ ]

A. package class_name B.class class_name

C. new class class_name D. new package class_name

15. University Question paper of previous years

Code No: 156CU **R18**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech III Year II Semester Examinations, August - 2022 SCRIPTING LANGUAGES

(Common to CSE, IT)

Time: 3 Hours **Max.Marks:75**

Answer any five questions All questions carry equal marks

- - -

1.a) Compare ruby language with Perl with respect to various features and list the advantages of Ruby language.
b) What is a cookie? Write the code in Ruby to demonstrate handling of cookies. [8+7]

2.a) What are web services and how do we use SOAP to implement web services in Ruby?
b) Describe different canvases available in Ruby-Tk. [8+7]

3.a) How do we embed a ruby language interpreter in C program?
b) Explain about Ruby type system. [7+8]

4.a) Write the syntax and explain the usage of different memory allocation functions available in Ruby.
b) Explain about global variables concept in Ruby with an example. [8+7]

5.a) How do we create a list and tuple in Perl? Demonstrate with a suitable coding example.
b) Discuss about defining regular expressions in Perl with examples. [8+7]

6.a) Enumerate various usages of Scripting languages.
b) Discuss about different conditional and iterative statements available in Perl with their syntaxes. [7+8]

7.a) Explain about pack and unpack commands in Perl with examples.
b) Write the steps for interfacing with operating system in Perl. [8+7]

8.a) Describe the control flow structures available in TCL.
b) Explain about event binding in Tk applications. [8+7]

---oo0oo---

Code No: 156CU **R18**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD B. Tech III Year II Semester Examinations, February - 2023 SCRIPTING LANGUAGES (Common to CSE, IT)

Time: 3 Hours **Max. Marks: 75**

Note: i) Question paper consists of Part A, Part B.
ii) Part A is compulsory, which carries 25 marks. In Part A, Answer all questions.
iii) In Part B, Answer any one question from each unit. Each question carries 10 marks and may have a, b as sub questions.

PART – A **(25 Marks)**

1.a) What do you mean by an iterator in Ruby? [2]
b) State the importance of RubyGems. [3]
c) Write about Fixnum immediate value in Ruby. [2]
d) Write an extension code in C that is plug-compatible with that Ruby class. [3]
e) Define the term Web scripting. [2]
f) How does chop( ) and chomp( ) functions in PERL differ? [3]
g) efine a module and a package. [2]
h) Give a brief file system operations in PERL. [3]
i) How to create a list in TCL? Give an example. [2]
j) What is the purpose of ‘exec’ and ‘append’ command in Tcl? [3]

PART – B **(50 Marks)**

2.a) “Embedding Ruby in HTML is a very powerful concept”. Support this statement using code snippets.
b) How to access Tk features from Ruby? Discuss briefly creating a simple application. [5+5]

OR

3. How Ruby CGI handles cookies and sessions? Demonstrate using the sample Ruby CGI Scripts. [10]

4. Explain various C/Ruby data type conversion functions and macros with clear syntax and illustrative examples for each. [10]

OR

5.a) Explain StringValue and StringValuePtr macros in Ruby with sample Ruby script.

b) How to embed ruby in your application? What is the role of Ruby interpreter? Explain. [4+6]

6. Explain various regular expressions and pattern matching operators available in PERL with examples. [10]

OR

7.a) What are the uses of scripting languages?
b) Explain various string and numeric built-in functions available in PERL. [4+6]

8.a) Outline the security issues in internet programming of PERL.
b)Give a detail note on using PERL interfacing to an operating system. [5+5]

OR

9.a) How to create packages and modules in PERL? Describe in detail.
b) What is the need of pack( ) and unpack( ) functions in Perl. [6+4]

10. What are the major functionalities of a Tcl Parser? Explain in detail about namespaces and recursive procedures in TCL. [10]

OR

11.a) Write Tk script to create a login form.
b) How are events handled in Tk? [5+5]

---ooOoo---

16. Unit-wise quiz questions with blooms

mapping Unit-I

Questions	Blooms
Explain the key features and advantages of Ruby as a scripting language. Provide examples to support your answer.	Understanding
Discuss the structure and execution process of Ruby programs, including the role of the interpreter and the various components involved.	Understanding
Explain the significance of package management in Ruby and how RUBYGEMS facilitates it. Discuss the benefits of using RUBYGEMS in a Ruby project.	Understanding
Describe the process of writing CGI scripts in Ruby for web development. Include the steps involved and highlight any important considerations.	Applying
Discuss the concept of cookies in web development and their implementation in Ruby. Explain how cookies can be used to maintain state and enhance user experience.	Understanding
Compare and contrast different choices of webservers for Ruby applications, such as Apache, Nginx, and WEBrick. Discuss their features, performance, and suitability for different scenarios.	Analyzing
Explain the role of SOAP and web services in Ruby web development. Discuss the advantages of using SOAP for intercommunication between different systems.	Understanding
Describe the basics of RubyTk and its usage in developing simple Tk applications. Provide examples of Tk applications developed using RubyTk.	Understanding
Discuss the different types of widgets available in RubyTk and their purpose. Explain how they can be used to create interactive and user-friendly GUI applications.	Understanding
Explain the concept of event binding in RubyTk and how it enables the response to user actions. Provide examples to illustrate the usage of event binding.	Applying
Explore the capabilities of the Canvas widget in RubyTk and how it can be used for drawing and graphical representation. Provide examples of complex graphical applications developed using RubyTk's Canvas.	Creating
Discuss the implementation of scrolling functionality in RubyTk applications. Explain different approaches to implement scrolling and their advantages/disadvantages.	Analyzing

Unit-II

Questions	Blooms
Discuss the process of extending Ruby using C. Explain the steps involved and provide examples to illustrate the concept of extending Ruby with C code.	Understanding
Explore the Jukebox extension as an example of extending Ruby. Discuss its purpose, features, and how it enhances the functionality of Ruby.	Understanding
Explain the memory allocation process in Ruby. Discuss different memory management techniques employed by Ruby and their impact on performance.	Understanding
Describe the Ruby type system and its key features. Discuss the different data types and type conversion mechanisms in Ruby, highlighting their importance in programming.	Understanding
Discuss the concept of embedding Ruby into other programming languages. Explain the benefits and use cases of embedding Ruby into languages like C, Java, or Python.	Understanding
Explore the process of embedding a Ruby interpreter into a C program. Discuss the steps involved and how it enables the execution of Ruby code within the C program.	Applying
Discuss the challenges and considerations involved in embedding a Ruby interpreter. Explain how to handle the interaction between Ruby and the host language, including variable passing and error handling.	Analyzing
Explain the concept of Ruby bindings and how they facilitate the integration of Ruby with other languages. Discuss the different types of bindings and their usage.	Understanding
Discuss the advantages and disadvantages of embedding Ruby compared to extending it using C. Compare the use cases and scenarios where each approach is more suitable.	Analyzing
Explore the concept of interoperability between Ruby and other languages. Discuss techniques such as using foreign function interfaces (FFI) or creating language-specific APIs to enable seamless communication between Ruby and other languages.	Understanding
Discuss the challenges and considerations involved in debugging and testing embedded Ruby code. Explain techniques and tools available for effective debugging and testing in an embedded Ruby environment.	Analyzing
Explain the concept of sandboxing Ruby code and its importance in embedded scenarios. Discuss techniques and best practices for ensuring security and isolation when executing embedded Ruby code.	Creating

Unit-III

Questions	**Blooms**
Discuss the concept of scripting and its origin. Explain the difference between scripts and programs, and the characteristics that define scripting languages.	Understanding
Explore the uses of scripting languages in various domains and industries. Discuss their advantages and limitations in comparison to other programming languages.	Analyzing
Explain the significance of web scripting and its role in developing dynamic and interactive web applications. Discuss the impact of scripting languages on web development.	Understanding
Provide an introduction to Perl as a scripting language. Discuss its features, strengths, and areas of application.	Understanding
explain the concept of names and values in Perl. Discuss variable declaration, initialization, and scoping in Perl.	Understanding
Discuss scalar expressions in Perl. Explain the operators and functions used to manipulate scalar values in Perl.	Applying
Describe the control structures available in Perl. Discuss conditional statements (if, else, elsif), loops (for, foreach, while), and control flow (next, last) in Perl.	Understanding
Explain arrays, lists, and hashes in Perl. Discuss their usage, manipulation, and how they store and organize data in Perl programs.	Understanding
Discuss strings in Perl, including string manipulation functions and regular expressions for pattern matching and text processing.	Understanding
Explain the concept of subroutines in Perl. Discuss their purpose, declaration, and usage in organizing code and promoting reusability.	Understanding
Discuss the importance and usage of pattern matching and regular expressions in Perl. Explain their syntax and application in text processing and data validation.	Applying
Explore the role of Perl in modern scripting languages. Discuss its popularity, community support, and how it fits into the broader universe of scripting languages.	Evaluating

Unit-IV

Questions	Blooms
Discuss the finer points of looping in Perl. Explain the different loop constructs available and their appropriate usage in different scenarios.	Understanding & Applying
Explain the concepts of pack and unpack in Perl. Discuss their purpose, syntax, and how they can be used for data manipulation and conversion.	Understanding & Applying
Discuss the filesystem operations in Perl. Explain how Perl interacts with the filesystem, including file handling, directory operations, and file permissions.	Understanding & Applying
Explain the concept of eval in Perl. Discuss its usage for dynamic code execution and error handling.	Understanding & Applying
Explore the data structures available in Perl, such as arrays, hashes, and nested data structures. Discuss their features, advantages, and appropriate use cases.	Understanding & Analyzing
Discuss the concepts of packages and modules in Perl. Explain how they help organize code, encapsulate functionality, and facilitate code reuse.	Understanding & Applying
Explain the concept of objects and object-oriented programming (OOP) in Perl. Discuss how Perl supports OOP principles and the benefits of using objects	Understanding & Applying
Discuss the techniques for interfacing Perl with the operating system. Explain how Perl can interact with system commands, access environment variables, and manipulate system processes.	Understanding & Applying
Discuss the creation of Internet-aware applications in Perl. Explain how Perl can be used for web development, including handling HTTP requests, processing forms, and generating dynamic web content.	Understanding & Applying
Explore the challenges and considerations involved in Internet programming using Perl. Discuss security issues, such as input validation, data sanitization, and protecting against common vulnerabilities.	Analyzing & Evaluating
Explain the importance of security in Perl applications. Discuss techniques and best practices for securing Perl code, handling sensitive data, and preventing common security vulnerabilities.	Evaluating & Applying
Discuss the concept of dirty hands Internet programming in Perl. Explain the practical aspects of working with network protocols, sockets, and client-server communication using Perl.	Evaluating & Applying

Unit-V

Questions	Blooms
Explain the structure and syntax of TCL. Discuss the key elements and components that make up a TCL script.	Remembering & Understanding
Discuss variables and data in TCL. Explain the different variable types, variable scope, and how data is stored and manipulated in TCL.	Understanding & Applying
Describe the control flow constructs in TCL. Discuss conditional statements (if-else), loops (while, for), and how they control the flow of execution in TCL programs.	Understanding & Applying
Explore the data structures available in TCL. Discuss arrays, lists, dictionaries, and their usage in storing and organizing data in TCL.	Understanding & Applying
Explain input/output operations in TCL. Discuss standard input/output, file input/output, and how TCL handles reading and writing data to files.	Understanding & Applying
Discuss procedures in TCL. Explain the concept of defining and using procedures, passing parameters, and returning values in TCL.	Understanding & Applying
Explore strings and pattern matching in TCL. Discuss string manipulation, regular expressions, and their usage in pattern matching and text processing in TCL.	Understanding & Applying
Explain file handling in TCL. Discuss file operations, such as opening, closing, reading, and writing files in TCL.	Understanding & Applying
Discuss advanced TCL concepts, such as eval, source, exec, and uplevel commands. Explain their usage and how they extend the capabilities of TCL	Understanding & Applying
Explain the concept of namespaces in TCL. Discuss how namespaces provide encapsulation, prevent naming conflicts, and enhance modularity in TCL programs.	Understanding & Applying
Discuss error handling and trapping errors in TCL. Explain techniques for handling and reporting errors, including try-catch blocks and error handling mechanisms in TCL.	Understanding & Applying
Explore event-driven programming in TCL. Discuss the event loop, event handling, and how to create event-driven applications in TCL.	Applying & Analyzing
Explain how to make TCL applications internet-aware. Discuss techniques for handling network protocols, interacting with servers, and creating web applications in TCL.	Applying & Analyzing
Discuss security issues in TCL programming. Explain common security vulnerabilities, techniques for securing TCL applications, and handling user input validation.	E valuating & Applying

explain the C interface in TCL. Discuss how TCL can be integrated with C code, including calling C functions from TCL and passing data between C and TCL.	Understanding & Applying

17. Tutorial Problems with blooms mapping

Bloom's Level	Tutorial Problem
Remembering	Write simple Ruby scripts (e.g., Hello World, prime numbers
Understanding	Explain Ruby script execution, package management with RubyGems
Applying	Create basic Ruby on Rails apps, CGI scripts, and RubyTk apps
Analyzing	Debug Ruby programs, compare web servers, analyze event handling in RubyTk
Evaluating	Evaluate Ruby code for errors, analyze security risks in CGI scripts, suggest improvements to UI in RubyTk
Creating	Build complex Ruby on Rails apps, web services, and interactive RubyTk applications
Remembering	Create basic Ruby C extensions, describe Ruby types and how they work with C.
Understanding	Explain C extension structure, embedding Ruby in C, handling Ruby types in C
Applying	Build simple and advanced C extensions, embed Ruby in other languages.
Analyzing	Debug C extensions, evaluate memory allocation, and type errors.
Creating	Build complex C extensions, custom Ruby types, or embed Ruby for dynamic behavior.
Evaluating	Review performance and security concerns in Ruby extensions.
Remembering	Define Perl variables, explain scripting language characteristics, list types of Perl variables.
Understanding	Explain the origin and uses of scripting languages, manipulate strings, control structures, and arrays in Perl.
Applying	Write Perl scripts for user input, control flow, arrays, subroutines, and regular expressions.
Analyzing	Analyze Perl code, compare Perl with other languages, optimize regular expressions and subroutines.
Evaluating	Evaluate Perl scripts for efficiency, performance, and readability.
Creating	Build modular, complex Perl programs using arrays, hashes, subroutines, and regular expressions.
Remembering	List looping constructs, system functions, and Perl modules for web applications.
Understanding	Explain packages, eval, looping structures, and system calls.

Applying	Create scripts using looping, pack/unpack, system functions, and web scrapers.
Analyzing	Analyze code for performance, security risks, and optimizations.
Evaluating	Evaluate the performance, security, and efficiency of Perl scripts.
Creating	Build complex Perl applications for system management, web services, and secure internet applications.
Remembering	List TCL syntax, commands, widgets, and modules for internet programming.
Understanding	Explain control flow, variables, events, and basic GUI features
Applying	Create scripts using input/output, loops, error handling, and Tk GUI.
Analyzing	Analyze code for performance, data structure usage, and event handling.
Evaluating	Evaluate code for efficiency, security risks, and design flaws.
Creating	Build modular, robust applications, integrating web and GUI features.

18. Assignment questions with blooms mapping

Unit-I

Blooms Mapping	Assignment Questions
Understand	Explain the concept of **binding events** in RubyTk. How are events like button clicks and key presses handled in RubyTk?
Understand	Explain the structure of a Ruby program. How does Ruby handle variables and data types?
Analyze	Analyze the MVC architecture in Ruby on Rails. Explain how controllers, models, and views work together to handle HTTP requests and responses.
Remember	What is the role of CGI scripts in Ruby web development? How are they used to handle HTTP requests and generate dynamic content?
Analyze	Analyze the different choices of web servers available for Ruby applications. What are the key differences between Apache, Nginx, and Puma in the Ruby ecosystem?
Evaluate	Evaluate the use of SOAP web services in Ruby. What are the pros and cons of using SOAP compared to RESTful services in web applications?
Analyze	Analyze the role of **Canvas** widgets in RubyTk. How do they differ from other widgets like labels and buttons?
Understand	Explain the process of installing and managing Ruby packages with **RubyGems**. How can you install, update, and remove gems?

Unit-II

Blooms Mapping	Assignment Questions
Understand	Explain how Ruby objects are created and manipulated in C. How do you allocate and assign values to Ruby objects within a C extension?
Understand	Describe how the **Jukebox extension** can be used to interact with music files in Ruby. How does it handle audio playback and metadata retrieval?

Remember	What is Ruby's garbage collection mechanism, and how does it manage memory allocation for Ruby objects?
Apply	Analyze how Ruby handles type conversion and coercion. Provide examples of implicit and explicit type conversions in Ruby.
Remember	What does it mean to **embed Ruby** in another language? Name a few scenarios where embedding Ruby might be beneficial
Understand	Explain how memory is allocated for objects in Ruby. How does Ruby's memory management system work when an object is created and later discarded?

Unit-III

Blooms Mapping	**Assignment Questions**
Remember	What is the origin of scripting languages? Provide a brief history of their development
Remember	List the different types of variables in Perl. What are the key differences between scalars, arrays, and hashes?
Understand	Explain how conditional statements like if, elsif, and else work in Perl. Provide an example where these structures are used.
Understand	Explain how arrays and hashes are used in Perl. How do you access an element in an array versus a hash?
Remember	What are regular expressions in Perl, and how do they help in pattern matching? Provide an example of a simple regular expression.
Analyze	Analyze the advantages of using regular expressions in Perl for text processing. What are the challenges one might face when using regular expressions for complex patterns?
Remember	What is a subroutine in Perl? How do you define and call a subroutine in Perl?

Unit-IV

Blooms Mapping	**Assignment Questions**
Understand	Explain how the last, next, and redo statements work in Perl loops. Provide an example for each.

Analyze	Analyze the advantages and disadvantages of using foreach loops versus while loops for iterating over arrays in Perl. When might one be preferred over the other?
Remember	What does the pack and unpack functions do in Perl? Give an example of how each is used
Understand	Explain how to open, read, and write to files in Perl. What are the different file modes in Perl?
Remember	What is the purpose of the eval function in Perl? How does it differ from require and use?
Understand	Explain how eval can be used for exception handling in Perl. How do you handle errors within eval blocks?
Understand	Explain how to create and access multi-dimensional arrays and hashes in Perl. Provide an example.
Remember	What is the difference between a **package** and a **module** in Perl? How do they help organize code?
Remember	What is object-oriented programming (OOP) in Perl? Name some key concepts related to OOP in Perl, such as classes, objects, and inheritance.
Understand	Explain how the system and exec functions work in Perl for executing external commands.
Understand	Explain the concept of **taint checking** in Perl. How does it help mitigate security risks?

Unit-V

Blooms Mapping	**Assignment Questions**
Remember	What is the basic structure of a TCL program? Describe its syntax and how it is different from other programming languages like C or Python.
Analyze	Analyze the differences between arrays and lists in TCL. How are they used, and how do they differ in terms of accessing and modifying elements?
Understand	Explain how input and output operations work in TCL. What is the role of the gets and puts commands, and how are they used to interact with the user?
Remember	What are the eval, source, exec, and uplevel commands in TCL? Briefly describe the purpose of each.

Understand	Explain the concept of trapping errors in TCL. How does the catch command help in error handling?
Apply	Write a TCL script that creates a simple GUI with a button. When the button is clicked, it prints "Hello, TCL!" to the console.
Apply	Write a TCL script that sends an HTTP GET request to a web server and prints the response.
Remember	What are some common security issues when writing TCL scripts, particularly in internet-aware applications?
Apply	Write a simple Tk application that creates a window with a button. When the button is clicked, the window changes color.
Apply	Write a Perl script using Perl-Tk that creates a simple window with a label displaying "Hello, Perl-Tk!"

19. List of Student's

21C31A6601	ABUL FATAH MOHAMMED AFFANULLAH
21C31A6602	ADDAGUDURU KRISHNA KOUSHIK
21C31A6603	ALUGOJU SARASWATHI
21C31A6604	AMANCHA VIVEK
21C31A6605	ANNARAPU VINAY
21C31A6606	ARENDRA DHARANI
21C31A6607	BAJJURI MAMATHA
21C31A6608	BHUKYA THRISHA
21C31A6609	BODDULA RAHUL
21C31A6610	BOLLA USHASREE
21C31A6611	BOLLOJU ANUSHA
21C31A6612	BUSIREDDY MADHURI
21C31A6613	CHALLURI DINESH KUMAR
21C31A6614	CHALLURI RAHUL
21C31A6615	CHELPURI RADHIKA
21C31A6616	CHITTHALURI ANUDEEP
21C31A6617	DEEKONDA CHANDU
21C31A6618	DENABOINA PRAVALIKA
21C31A6619	DUPATI SRICHARAN
21C31A6620	ENAGANTI MANOJ
21C31A6621	FAISAL SYED
21C31A6622	GANDI GOUTHAM
21C31A6623	GATTIKOPPULA AJAY
21C31A6624	GUDURU SAI RAJ
21C31A6625	GUMMADIRAJU REVATHI
21C31A6626	JADALA RAM SAGAR
21C31A6627	KALLEPELLY ARCHANA
21C31A6628	KOMURAVELLI SHIVA KUMAR
21C31A6629	KORRA KAVYA
21C31A6630	KOYYADA CHANDAN RAJ
21C31A6631	KUCHANA RACHANA
21C31A6632	KUKKALA RAVI KIRAN
21C31A6633	KYATHAM ROHITH
21C31A6634	LADE KAVYASRI
21C31A6635	LAKAVATH VENKANNA
21C31A6636	MADIPELLY MUKTHA NANDHINI
21C31A6637	MOHAMMED ABDUL RAHAMAN
21C31A6638	MOHAMMED RAJJU
21C31A6639	MOHAMMED SAMEER
21C31A6640	MOHAMMED YAKUB FARAZ KHAN
21C31A6641	MUNIGALA POORNACHANDER

21C31A6642	MUNIGANTI AKHIL
21C31A6643	NALLA ADITHYA
21C31A6644	NALLA LAXMI PRASANNA
21C31A6645	NARUGULA RAKESH
21C31A6646	NAVEEN ADEPU
21C31A6647	NUNAVATH BALARAJU
21C31A6648	PARUNANDHI PAVAN WESLY
21C31A6649	PILLALAMARRI SUDHEER
21C31A6650	PILLI HARSHASRI
21C31A6651	PITTA ARAVIND
21C31A6652	SABBANI RAKSHITHA
21C31A6653	SHAIK SALMAN
21C31A6654	SHANIGARAPU JHANSY
21C31A6655	SILUVERU PRINCE
21C31A6656	SINGARAPU SRAVANI
21C31A6657	SRIRAMULA SRILEKHA
21C31A6658	TAKKALLAPALLY KANISHKA
21C31A6659	THADAKA SRI POOJA
21C31A6660	THATIKONDA NARESH
21C31A6661	VALLAKATLA TEJA
21C31A6662	VELPURI NIHARIKA
21C31A6663	YARRAM SAI DATH
22C35A6601	AKULA ROHITH
22C35A6602	CHINDAM SHIVA KUMAR
22C35A6603	GADE SUSHMA SRI
22C35A6604	MOHAMMAD AVEZ
22C35A6605	MADHUKAR
22C35A6606	RAVIRAKULA SANDEEP

20. Scheme and solution of internal tests.

For theory subjects, during a semester, there shall be two mid-term examinations.

Each Mid-Term examination consists of two parts i) Part – A for 10 marks, ii) Part – B for 20 marks with a total duration of 2 hours as follows:

1. Mid Term Examination for 30 marks:

a. Part - A : Objective/quiz paper for 10 marks.

b. Part - B : Descriptive paper for 20 marks.

The objective/quiz paper is set with multiple choice, fill-in the blanks and match the following type of questions for a total of 10 marks. The descriptive paper shall contain 6 full questions out of which, the student has to answer 4 questions, each carrying 5 marks. The average of the two Mid Term Examinations shall be taken as the final marks for Mid Term Examination (for 30 marks). The remaining 10 marks of Continuous Internal Evaluation are distributed as:

2. Assignment for 5 marks. (Average of 2 Assignments each for 5 marks)

3. Subject Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the concerned subject for 5 marks.

While the first mid-term examination shall be conducted on 50% of the syllabus, the second mid-term examination shall be conducted on the remaining 50% of the syllabus.

Five (5) marks are allocated for assignments (as specified by the subject teacher concerned). The first assignment should be submitted before the conduct of the first mid-term examination, and the second assignment should be submitted before the conduct of the second mid-term examination. The average of the two assignments shall be taken as the final marks for assignment (for 5 marks).

Subject Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the subject concerned for 5 marks before II Mid-Term Examination.

□ The Student, in each subject, shall have to earn 35% of marks (i.e. 14 marks out of 40 marks) in CIE, 35% of marks (i.e. 21 marks out of 60) in SEE and Over all 40% of marks (i.e. 40 marks out of 100 marks) both CIE and SEE marks put together.

The student is eligible to write Semester End Examination of the concerned subject, if the student scores ≥ 35% (14 marks) of 40 Continuous Internal Examination (CIE) marks.

21.Marks sheet.

Q.No.	Answer any two questions.	Marks	Level of Bloom Taxonomy	CO
1	What is SOAP?Explain about SOAP Structure with an Example?	5	Analyze	CO1
2	What is Tk-Widgets?Write Sample Tk Applications?	5	Understand	CO2
3	Discuss the various characteristics of a Scripting Languages?	5	Remember	CO2

Sl. No.	Roll Number	Name of the Candidate	Marks Awarded			Part - B (Theory-10)	Part - A (Quiz-10)	A+B=20	Unit Test(5)	Assessment(5)	Grand Total (30)
			Q1	Q2	Q3						
1	21C31A6601	ABUL FATAH MOHAMMED AFFANULLAH	4	4		8	10	18	5	4	27
2	21C31A6602	ADDAGUDURU KRISHNA KOUSHIK	5	4		9	9	18	5	3	26
3	21C31A6603	ALUGOJU SARASWATHI	4	4		8	10	18	5	5	28
4	21C31A6604	AMANCHA VIVEK	5	4		9	10	19	5	5	29
5	21C31A6605	ANNARAPU VINAY	4	5		9	8	17	5	4	26
6	21C31A6606	ARENDRA DHARANI	4	4		8	9	17	5	4	26
7	21C31A6607	BAJJURI MAMATHA	5	4		9	9	18	5	5	28
8	21C31A6608	BHUKYA THRISHA	4	5		9	10	19	5	5	29
9	21C31A6609	BODDULA RAHUL	5	5		10	9	19	5	3	27
10	21C31A6610	BOLLA USHASREE	4	5		9	8	17	5	5	27
11	21C31A6611	BOLLOJU ANUSHA	4		4	8	7	15	5	3	23
12	21C31A6612	BUSIREDDY MADHURI	5		4	9	9	18	4	3	25
13	21C31A6613	CHALLURI DINESH KUMAR	4	3		7	7	14	4	3	21
14	21C31A6614	CHALLURI RAHUL	4	4		8	7	15	4	5	24
15	21C31A6615	CHELPURI RADHIKA	4	4		8	8	16	5	4	25
16	21C31A6616	CHITTHALURI ANUDEEP	4	4		8	8	16	5	5	26
17	21C31A6617	DEEKONDA CHANDU	4	4		8	9	17	5	5	27
18	21C31A6618	DENABOINA PRAVALIKA	5	4		9	9	18	4	5	27
19	21C31A6619	DUPATI SRICHARAN	4	4		8	8	16	4	4	24
20	21C31A6620	ENAGANTI MANOJ	3	4		7	8	15	4	0	19
21	21C31A6621	FAISAL SYED					AB	AB	4	0	4
22	21C31A6622	GANDI GOUTHAM				3	9	12	4	4	22
23	21C31A6623	GATTIKOPPULA AJAY	4			4	10	14	4	0	18
24	21C31A6624	GUDURU SAI RAJ	5	4		9	10	19	5	5	29
25	21C31A6625	GUMMADIRAJU REVATHI	5	4		9	9	18	5	5	28
26	21C31A6626	JADALA RAM SAGAR	5	4		9	9	18	4	5	27
27	21C31A6627	KALLEPELLY ARCHANA	4	4		8	10	18	4	5	27
28	21C31A6628	KOMURAVELLI SHIVA KUMAR	4	4		8	8	16	4	4	24
29	21C31A6629	KORRA KAVYA	4	5		9	8	17	4	5	26
30	21C31A6630	KOYYADA CHANDAN RAJ	3	4		7	9	16	4	3	23
31	21C31A6631	KUCHANA RACHANA	4	4		8	9	17	4	5	26
32	21C31A6632	KUKKALA RAVI KIRAN	4	4		8	8	16	5	3	24
33	21C31A6633	KYATHAM ROHITH	4	4		8	9	17	4	3	24
34	21C31A6634	LADE KAVYASRI	4	5		9	9	18	4	4	26
35	21C31A6635	LAKAVATH VENKANNA	3	4		7	8	15	AB	3	18
36	21C31A6636	MADIPELLY MUKTHA NANDHINI	5	4		9	10	19	5	5	29
37	21C31A6637	MOHAMMED ABDUL RAHAMAN	4	4		8	10	18	5	5	28
38	21C31A6638	MOHAMMED RAJJU	3	4		7	8	14	4	5	23
39	21C31A6639	MOHAMMED SAMEER	3	3		6	9	15	4	3	22
40	21C31A6640	MOHAMMED YAKUB FARAZ KHAN	4			4	9	13	4	3	21
41	21C31A6641	MUNIGALA POORNACHANDER	4			4	9	13	4	3	20
42	21C31A6642	MUNIGANTI AKHIL	3	3		6	9	15	4	4	23
43	21C31A6643	NALLA ADITHYA	4	4		8	9	17	4	5	26
44	21C31A6644	NALLA LAXMI PRASANNA	4	5		9	9	18	5	5	28
45	21C31A6645	NARUGULA RAKESH	4		4	8	9	17	5	3	25
46	21C31A6646	NAVEEN ADEPU	3	2		5	9	14	4	3	22
47	21C31A6647	NUNAVATH BALARAJU	4			4	9	13	4	3	19
48	21C31A6648	PARUNANDHI PAVAN WESLY	3			3	9	12	AB	3	15
49	21C31A6649	PILLALAMARRI SUDHEER	4	4		8	10	18	5	3	26
50	21C31A6650	PILLI HARSHASRI	4	3		7	9	16	4	5	25
51	21C31A6651	PITTA ARAVIND	3			3	10	15	5	4	24
52	21C31A6652	SABBANI RAKSHITHA	5	4		9	10	19	5	4	28
53	21C31A6653	SHAIK SALMAN	3	2		5	9	14	4	5	23
54	21C31A6654	SHANIGARAPU JHANSY	4	4		8	9	17	5	3	23
55	21C31A6655	SILUVERU PRINCE	4	3		7	9	16	AB	4	20
56	21C31A6656	SINGARAPU SRAVANI			5	5	10	18	5	5	28
57	21C31A6657	SRIRAMULA SRILEKHA	5	4		9	10	15	5	3	23
58	21C31A6658	TAKKALLAPALLY KANISHKA	3	3		6	10	19	5	5	29
59	21C31A6659	THADAKA SRI POOJA	4	3		7	10	16	5	3	24
60	21C31A6660	THATIKONDA NARESH	4	3		7	10	17	5	3	25
61	21C31A6661	VALLAKATLA TEJA	4	3		7	10	17	5	5	27
62	21C31A6662	VELPURI NIHARIKA	4	4		8	9	17	AB	3	20
63	21C31A6663	YARRAM SAI DATH	3	3		6	9	15	AB	3	18
64	22C35A6601	AKULA ROHITH	3	3		6	9	15	5	4	24
65	22C35A6602	CHINDAM SHIVA KUMAR	3	3		6	9	15	4	4	23
66	22C35A6603	GADE SUSHMA SRI	AB	AB	AB	AB	AB	AB	AB	0	0
67	22C35A6604	MOHAMMAD AVEZ	3	3		7	9	16	4	4	24
68	22C35A6605	MADHUKAR	3	3		6	6	12	4	4	20
69	22C35A6606	RAVIRAKULA SANDEEP						14	4	4	22

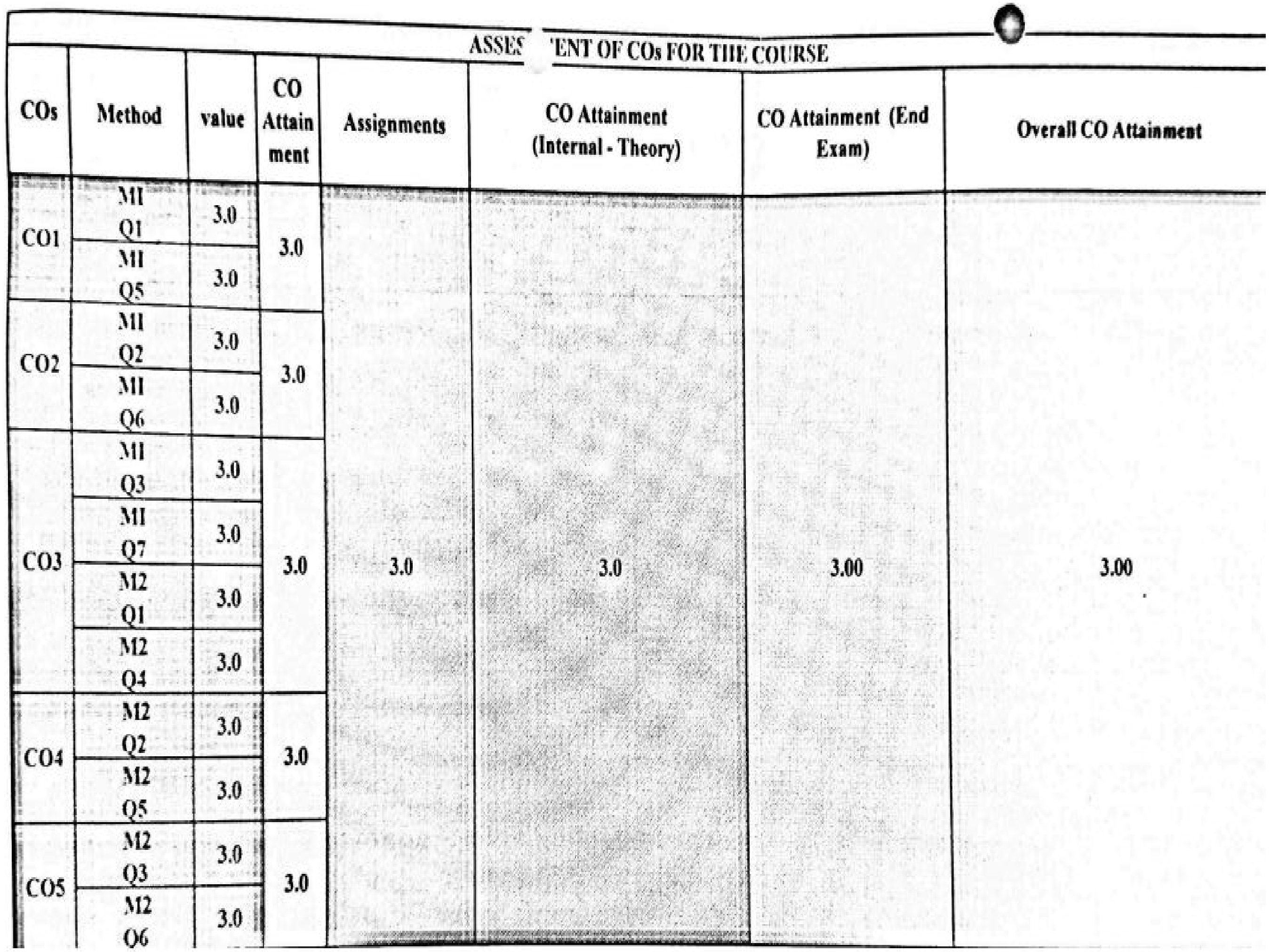
21. Result analysis for internal exams (tests) with respect to COs-Pos

Q.No.	Answer any two questions.	Marks	Level of Bloom Taxonomy	CO
1	Write about important terminologies in Neural Networks?	5	Understand	CO1
2	Discuss about Fixed Weight Competitive Nets with examples?	5	Understand	CO1
3	What is mean by deep learning? Write about Historical Trends in Deep learning?	5	Understand	CO2

Sl. No.	H.T. No	Marks Awarded			Part - B (Theory)	Part - A (Quiz)	Part - A + B	Unit Test - 1	Assessment	Grand Total
		Q1.	Q2.	Q3						
1	21C31A6601	5		5	10	10	**20**	5	**4**	**29**
2	21C31A6602	5		4	9	10	**19**	5	**4**	**28**
3	21C31A6603	5		5	10	10	**20**	5	**4**	**29**
4	21C31A6604	5	5	5	10	9	**19**	5	**5**	**29**
5	21C31A6605	5		5	10	10	**20**	5	**4**	**29**
6	21C31A6606	3		5	8	10	**18**	5	**4**	**27**
7	21C31A6607	5		5	10	9	**19**	5	**4**	**28**
8	21C31A6608	4		4	8	9	**17**	5	**4**	**26**
9	21C31A6609	5		5	10	10	**20**	5	**4**	**29**
10	21C31A6610	4		4	8	10	**18**	5	**4**	**27**
11	21C31A6611	4		2	6	10	**16**	5	**3**	**24**
12	21C31A6612	5		5	10	10	**20**	5	**4**	**29**
13	21C31A6613	4		3	7	9	**16**	5	**3**	**24**
14	21C31A6614			4	4	10	**14**	5	**4**	**23**
15	21C31A6615	4		4	**8**	**9**	**17**	5	**3**	**25**
16	21C31A6616	1		1	2	9	**11**	5	**4**	**20**
17	21C31A6617	5		5	10	10	**20**	5	**4**	**29**
18	21C31A6618	5		4	9	9	18	5	**4**	**27**
19	21C31A6619	3	4	4	**8**	**9**	**17**	5	**4**	**26**
20	21C31A6620	2			2	10	**12**	5	**4**	**21**
21	21C31A6621	ABSENT						5	**3**	**8**
22	21C31A6622	2			2	10	**12**	5	**4**	**21**
23	21C31A6623	2	2		**4**	**10**	**14**	5	**4**	**23**

24	21C31A6624	5		5	10	10	**20**	5	**4**	**29**
25	21C31A6625	5		5	10	10	**20**	5	4	**29**
26	21C31A6626	4		5	9	10	**19**	5	4	**29**
27	21C31A6627	5		5	10	10	**20**	5	4	**29**
28	21C31A6628					9	**9**	5	3	**17**
29	21C31A6629			4	4	9	**13**	5	4	**22**
30	21C31A6630	4		4	**8**	**9**	**17**	5	4	**26**
31	21C31A6631	5		5	10	10	**20**	5	4	**29**
32	21C31A6632	1		4	5	9	**14**	5	3	**22**
33	21C31A6633	3		3	6	10	**16**	5	4	**25**
34	21C31A6634	5		5	**10**	**9**	**19**	5	5	**24**
35	21C31A6635	5		4	9	10	**19**	5	3	**26**
36	21C31A6636	5		5	10	10	**20**	5	4	**29**
37	21C31A6637	4		4	8	9	**17**	5	4	**26**
38	21C31A6638	2	2		4	10	**14**	5	4	**23**
39	21C31A6639	4		2	6	9	**15**	5	4	**24**
40	21C31A6640	4			4	9	**13**	5	3	**21**
41	21C31A6641	5			5	9	**14**	5	4	**23**
42	21C31A6642	3		3	6	10	**16**	5	3	**23**
43	21C31A6643	4		5	9	9	**18**	5	4	**27**
44	21C31A6644	5		4	9	10	**19**	5	4	**28**
45	21C31A6645	2		4	6	9	**15**	5	4	**24**
46	21C31A6646			4	4	8	**12**	5	3	**20**
47	21C31A6647	3	3		6	9	**15**	5	4	**24**
48	21C31A6648	1		2	3	9	**12**	5	4	**16**
49	21C31A6649	5		5	10	10	**20**	5	**4**	**29**
50	21C31A6650	5		5	10	10	**20**	5	**4**	**29**
51	21C31A6651	2		3	5	9	**14**	5	**4**	**23**
52	21C31A6652	5		5	10	10	20	5	**5**	**30**
53	21C31A6653	2		3	5	10	**14**	5	**4**	**24**
54	21C31A6654	4		5	9	10	**19**	5	**4**	**28**
55	21C31A6655	4		4	8	10	**18**	5	**3**	**21**
56	21C31A6656	5		5	10	10	**20**	5	**4**	**29**
57	21C31A6657	4		3	7	10	**17**	5	**4**	**26**
58	21C31A6658	5		4	9	10	**19**	5	**4**	**28**
59	21C31A6659	5		4	9	9	**18**	5	**4**	**27**
60	21C31A6660	5		5	10	9	**19**	5	**4**	**28**
61	21C31A6661	5	4		9	10	**19**	5	4	28
62	21C31A6662	5		4	9	10	**19**	5	**5**	**24**
63	21C31A6663	5		4	9	10	**19**	5	**4**	**23**
64	22C35A6601	4		4	8	10	**18**	5	**4**	**27**
65	22C35A6602	3		3	6	10	16	5	**3**	**24**
66	22C35A6603	ABSENT						0	**0**	**0**
67	22C35A6604	5		4	9	10	19	5	**4**	**28**
68	22C35A6605	3		3	8	10	18	5	**3**	**24**
69	22C35A6606	2		3	5	10	15	5	**4**	**24**

22. CO and PO attainment sheet



ASSES 'ENT OF COs FOR THE COURSE

COs	Method	value	CO Attainment	Assignments	CO Attainment (Internal - Theory)	CO Attainment (End Exam)	Overall CO Attainment
CO1	M1 Q1	3.0	3.0	3.0	3.0	3.00	3.00
	M1 Q5	3.0					
CO2	M1 Q2	3.0	3.0				
	M1 Q6	3.0					
CO3	M1 Q3	3.0	3.0				
	M1 Q7	3.0					
	M2 Q1	3.0					
	M2 Q4	3.0					
CO4	M2 Q2	3.0	3.0				
	M2 Q5	3.0					
CO5	M2 Q3	3.0	3.0				
	M2 Q6	3.0					

26. References, Journals, websites and E-links if any

Websites: https://www.ruby-lang.org/en/documentation/quickstart/
https://www.tpointtech.com/ruby-tutorial https://www.tcl-lang.org/man/tcl8.5/tutorial/tcltutorial.html
https://www.perl.org/

Reference https://media.pragprog.com/articles/other-published-articles/ddj_ruby.pdf

https://books.google.co.in/books?hl=en&lr=&id=jcUbTcr5XWwC&oi=fnd&pg=PR5&dq=ruby+journal+papers&ots=fMEiAh8vaD&sig=AZz2tYfKNjkyyioujMt-JW1QAdg#v=onepage&q&f=false

https://www.google.co.in/books/edition/Win32_Perl_Programming/C6G0evV1ekwC?hl=en&gbpv=1&dq=perl+language++journal+papers&printsec=frontcover