<table>
<tr><td colspan="5">Devi Ahilya Vishwavidhyalaya, Indore, India
Institute of Engineering & Technology</td><td colspan="2">III Year B.Tech. (Electronics and Telecommunication Engineering)</td></tr>
<tr><td>Course Code & Name</td><td colspan="6">Instructions Hours per Semester and Credits</td></tr>
<tr><td rowspan="3">5RTPC4

SOFTWARE WORKSHOP II</td><td colspan="2">Classroom Instruction (CI)</td><td>Lab Instruction (LI)</td><td>Term Work (TW) and Self Learning (SL)</td><td>Total no. of Hours Per semester</td><td>Total Credits (Total Hours/30)</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>TW+SL</td><td rowspan="2">60</td><td rowspan="2">2</td></tr>
<tr><td>00</td><td>00</td><td>20</td><td>40</td></tr>
</table>

Course Learning Objectives:

- To enable developing Java based application programs using iteration statements, conditional statements, strings, arrays and vectors.
- To develop Java based application program involving the concepts of inheritance, interfaces and data abstraction.
- To understand developing programs involving user defined packages, creating GUIs based on Java AWT and Java Swing.
- To design Digital circuits on a Simulator and verify its working.

Prerequisites (if any): basic knowledge of computer programming

List of Practical Assignments

MODULE I

1. Write a program to check whether Java is installed in your system and it should give output as: Java version (ii) Java runtime version (iii) Java home directory (iv) Java vendor name.

2. Write a program in which user provides inputs and converts inches to centimeters.

3. Write a program in which the program gives output numbers 1 to 10 along with their square roots.

4. Write a program that takes user input for distance (in mts) and time (in sec, mt, hrs) and display the speed in metres per second, Km/hr and miles per hour.

5. Write an application that asks a user to enter an integer. Pass the integer to a method that determines whether the number is even and returns a Boolean value. Display a statement that indicates whether the integer is even or odd.

6. Write a program in which combinational Boolean operators are used.

7. Write a program in which the two numbers are taken from the user; the numbers are compared and the result is displayed.

8. Write a program that reads an integer between 0 and 1000 and adds all the digits in integer. (example if the input integer is 675, the output should be displayed as 17).

9. Write a program to find factorial of a given number which is provided by the user.

10. The numbers in the sequence

1 1 2 3 5 8 13 21… ..... are called as Fibonacci numbers. Write a program using do while loop to calculate and print the first m Fibonacci numbers.

11. Write a program in which a class Circle is defined. Following methods are defined within this class to compute and display circumference and arc length of a circle :

    (i) circumCircle() - to compute the circumference of a circle.

    (ii) arcLenght – to compute the lenght of the arc for a given angle.

Write a program to compute and display circumference and arc length when the radius is 10 and angle subtended is 37 degrees.

12. Write a program that prompts a user to input a positive integer. It should print the multiplication table of that number.
13. Write a program using arrays to find and display the multiplication of two matrices.

14. Write a program that displays the following patterns:

```
    1
   222
  33333
 4444444
555555555
```

```
    1
   212
  32123
 4321234
543212345
```

15. Write a program to find and display addition and subtraction of two 3X3 matrices.

16. Write a program in which a string is given; the program counts the characters and displays the result.

17. Write a program in which hierarchical inheritance is implemented to find and display the area and perimeter of circle in one of the class, in another subclass, area and perimeter of rectangle is found and displayed. In addition to this, cost of fencing the circular and

rectangular plot is also calculated and displayed.

18. Write a program in which method overriding is used in which sum of first 10 natural numbers and product of first 10 natural numbers is displayed.

19. Write a program using abstract class in which there are abstract methods to convert a given String value into int type and String value into double type.

20. Write a program in which multiple inheritance is implemented using interfaces in which class implements multiple interfaces.

21. Write a program in which multiple inheritance is implemented using interfaces in which an interface extends other interfaces.

22. Write a program in which user defined package is implemented and the class of this user defined package is imported in the application program.

23. Write a program in which user defined exception is implemented. In this program, the area of a room is calculated and the cost of white washing is also evaluated. The inputs regarding the parameters including length, breadth, and height of the room are taken through Command line. If these input parameters are below 1, then raise an exception; otherwise, calculate the area and cost and display the result.

24. Design an application for some fast food shop. It allows the user to make various fast food choices (e.g sandwitch etc.) from the list and the application displays the corresponding price. The user can choose from four different types of food options at different prices. It also displays the total price user has to pay as per the choices made by the user.

MODULE-2 Advance Digital Design

Assignment 1: Design of a 16-to-4 Priority Encoder

Design and implement a 16-to-4 Priority Encoder that identifies the highest-priority active input and generates its corresponding binary output along with a valid output signal. Verify the circuit for multiple input combinations, including cases where multiple inputs are active simultaneously. Analyze the importance of priority encoders in interrupt handling, resource allocation, and communication systems.

Assignment 2: Design of an 8-bit Arithmetic Logic Unit (ALU)

Design an 8-bit Arithmetic Logic Unit capable of performing arithmetic operations such as addition, subtraction, increment, and decrement, along with logical operations including AND, OR, XOR, XNOR, and NOT. Use control signals to select the desired operation and verify the functionality through simulation. Evaluate the role of the ALU in processor architectures and digital signal processing applications.

Assignment 3: Design of an 8-bit Barrel Shifter

Design an 8-bit Barrel Shifter capable of performing logical left shift, logical right shift, rotate left, and rotate right operations. Implement the design using multiplexers and

control signals so that shifting is completed in a single operation. Simulate the circuit for different shift amounts and compare its performance with conventional sequential shifting techniques.

Assignment 4: Design of a 4-bit Carry Look-Ahead Adder

Design a 4-bit Carry Look-Ahead Adder using generate and propagate logic to reduce carry propagation delay. Compare its performance with a Ripple Carry Adder in terms of propagation delay, hardware complexity, and speed. Verify the design using different test cases and discuss its applications in high-speed arithmetic units.

Assignment 5: Design of a Universal Shift Register

Design a 4-bit Universal Shift Register capable of performing Serial-In Serial-Out (SISO), Serial-In Parallel-Out (SIPO), Parallel-In Serial-Out (PISO), and Parallel-In Parallel-Out (PIPO) operations. Include left shift, right shift, parallel load, and hold operations using suitable control inputs. Simulate each operating mode and explain the applications of shift registers in communication and data processing systems.

Assignment 6: Design of a Synchronous MOD-12 Counter

Design a synchronous MOD-12 counter using JK or D flip-flops and suitable combinational logic. Verify the counting sequence from 0 to 11 and ensure that the counter resets correctly after reaching the terminal count. Analyze the advantages of synchronous counters over asynchronous counters and discuss their applications in digital systems.

Assignment 7: Design of a Johnson Counter

Design a 4-bit Johnson Counter using D flip-flops and simulate its counting sequence. Observe the unique output patterns generated during each clock cycle and verify the operation through timing waveforms. Discuss the advantages of Johnson counters over ring counters and identify practical applications in sequence generation and timing circuits.

Assignment 8: Design of a Serial Adder

Design a Serial Adder using a shift register, full adder, and flip-flops to perform binary addition one bit at a time. Simulate the circuit for different input combinations and compare its hardware requirements and execution time with those of a parallel adder. Explain the applications of serial arithmetic in low-cost and resource-constrained digital systems.

Assignment 9: Design of an 8×4 Register File

Design an 8×4 Register File consisting of eight 4-bit registers with one write port and two read ports. Implement the required decoding and multiplexing logic to perform read and write operations. Verify correct storage and retrieval of data using simulation and discuss the role of register files in processor datapaths and embedded systems.

Assignment 10: Design of a 16×8 Single-Port RAM

Design and implement a 16×8 Single-Port RAM supporting synchronous write and asynchronous or synchronous read operations. Verify the memory by storing and retrieving data from different memory locations. Analyze the importance of RAM organization, memory addressing, and control signals in digital computing systems.

Recommended Online Simulators

- CircuitVerse (Graphical digital logic simulator)

Course Outcomes:

CO.No.	CO
CO1	Develop Java application programs using variables, operators, control statements, methods, arrays, strings, matrices, and object-oriented programming concepts to solve computational problems.
CO2	Design and implement Java programs using inheritance, method overriding, abstract classes, interfaces, packages, exception handling, and GUI components to develop modular and reusable software applications.
CO3	Design and simulate medium-complexity combinational digital circuits such as Priority Encoders, Arithmetic Logic Units (ALU), Barrel Shifters, and Carry Look-Ahead Adders using HDL and online simulation tools.
CO4	Design and verify sequential digital circuits including Universal Shift Registers, Synchronous Counters, Johnson Counters, Serial Adders, Register Files, and Single-Port RAM using HDL and digital simulation platforms.
CO5	Analyze, verify, and document software and digital hardware designs by interpreting simulation results, validating functionality through test cases, and utilizing modern engineering tools for implementation and debugging.

CO-PO-PSO Relationship

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
5RTPC4.CO1	**3**	**2**	**2**		**2**						**1**	2	1	2
5RTPC4.CO2	**3**	**2**	**3**	**1**	**2**				**1**	**1**	**1**	2	2	2
5RTPC4.CO3	**3**	**3**	**3**	**2**	**3**						**1**	2	3	1
5RTPC4.CO4	**3**	**3**	**3**	**2**	**3**						**1**	2	3	2
5RTPC4.CO5	**2**	**3**	**2**	**3**	**3**			**1**	**1**	**1**	**2**	1	2	3