Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

Reg. Number: 23BRS1009

Continuous Assessment Test (CAT) – I AUGUST 2025

Programme	:	B. Tech AI & R	Semester	:	Fall Semester 2025-26
Course Code & Course Title	:	BCSE423L Robot Programming	Class Number	:	CH2025260101554 CH2025260101641 CH2025260101642
Faculty	:	51679 - Dr. GANALA SANTOSHI 53039 - Dr. SUVIDHA RUPESH KUMAR 53623 - Dr. SARANYARAJ D	Slot	:	A2
Duration	:	90 minutes	Max. Mark		50

General Instructions:

- Write only your registration number on the question paper in the box provided and do not write oth information
- Use statistical tables supplied from the exam cell as necessary
- Use graph sheets supplied from the exam cell as necessary
- Only non-programmable calculator without storage is permitted

Answer all questions

Q. No	Sub Sec.	Description	Marks	CO	BT Level
1	a	List the primary operational domains used to classify robotic systems based on their intended application and environment. (5 marks)	10	1	
	b	Given the following systems, identify their classification category from the domains listed above. Provide a brief justification for each. (3 marks) 1. A remotely operated underwater inspection device used for pipeline maintenance 2. An autonomous surgical assistant system for minimally invasive procedures 3. A warehouse inventory scanner mounted on a mobile platform			
	c	Design a conceptual layout for a compact autonomous monitoring unit, clearly specifying the essential structural subsystems and embedded control elements required for operation. (2 marks)			
2	a	In designing a ROS workspace for a multi-module robotic system, how would you structure your catkin workspace and ROS packages to efficiently manage inter-package dependencies, build order, and namespace collisions? (5 marks)	10	2	
	b	Explain, how you would utilize package.xml and CMakeLists.txt configurations to handle complex dependency graphs, and discuss strategies to optimize build times and facilitate continuous integration. (5 marks)			
		A ROS service server node that listens on the `/check_fire_alert` service. It accepts requests containing: • A zone identifier (string) • The current temperature reading (float) • A smoke detection flag (boolean) The server evaluates the temperature and smoke status using predefined criteria (temperature > 45.0°C and smoke detected). If both conditions are met, it responds with a fire alert message; otherwise, it responds indicating that the zone is normal. A custom service definition file named `FireAlert.srv` is	15	2	

CODE:

```
#!/usr/bin/env python3
import rospy
from forest_monitoring.srv import FireAlert, FireAlertResponse
def handle_fire_alert(req):
    if req.temperature > 45.0 and req.smoke_detected:
        message = f"ALERT: Possible fire at {req.zone_id}"
    else:
        message = f"Zone {req.zone_id} is normal"
    rospy.loginfo(message)
    return FireAlertResponse(message)
def fire_alert_server():
    rospy.init_node('fire_alert_server')
    service = rospy.Service('check_fire_alert', FireAlert,
handle_fire_alert)
    rospy.loginfo("Service 'check_fire_alert' is available.")
    rospy.spin()
if __name__ == "__main__":
    fire_alert_server()
```

a

Answer the following questions: (5 marks)

1. What is the role of the `handle_fire_alert` function? (2 marks)
2. What is the purpose of `rospy.spin()` in this context? (1 mark)
3. Suggest one modification to make the fire alert evaluation more configurable (e.g., adjustable temperature threshold). (2 marks)

b

Write a ROS service client node in Python or C++ that performs the following: (5 marks)

1. Sends requests to the `/check_fire_alert` service.
2. Each request contains the zone ID (string), temperature (float), and smoke status (boolean).
3. Receives the response from the service and displays the alert message.

c

Service Definition and Build Configuration: (5 marks)

1. Write the full content of the `FireAlert.srv` file. (1 mark)
2. Describe the required modifications to integrate this service into your ROS package:
    - `package.xml` (2 marks)
    - CMakeLists.txt (2 marks)

Consider the following node:

```
#!/usr/bin/env python3
import rospy
from std_msgs.msg import Float32
def publish_sensor_data():
    pub = rospy.Publisher('sensor_data', Float32, queue_size=10)
    rospy.init_node('sensor_data_publisher', anonymous=True)
    rate = rospy.Rate(2)  # 2 Hz
    sensor_value = 25.0
    while not rospy.is_shutdown():
        rospy.loginfo(f"Publishing sensor value: {sensor_value}")
        pub.publish(sensor_value)
```

15

2

```
        sensor_value += 0.5  # Simulate changing sensor readings
        rate.sleep()
if __name__ == '__main__':
    try:
        publish_sensor_data()
    except rospy.ROSInterruptException:
        pass
```

a	Identify the node type with proper justification. (3 marks)
b	Write the corresponding node receiving the sensor_data and logs a warning if the value exceeds 30.0. (8 marks)
c	Why is queue_size=10 important in the publisher setup? What would happen if this value were too small or too large? (4 marks)

**********All the best **********