

Scoring Indicators

COURSE NAME: Object Oriented Programming

COURSECODE: 4131

QID: 2103230063


QNo	Scoring Indicators	Split score	SubTotal	Total score
<u>PART A</u>				9
I.1	Encapsulation	1	1	9
I.2	true	1	1	
I.3	extends	1	1	
I.4	abstract	1	1	
I.5	Layout Managers control the position of components within a container.	1	1	
I.6	1. Top-level containers. 2. Lightweight containers.	½ X 2	1	
I.7	primary key.	1	1	
I.8	A DBMS (Database Management System) is a software system that is used to create, manage, and maintain databases.	1	1	
I.9	getConnection()	1	1	
<u>PART B</u>				24
II.1	Java Virtual Machine(JVM) is a program which provides the runtime environment to execute Java programs. JVM is a virtual machine that resides in the real machine and the machine language for JVM is byte code. The Java compiler generates byte code for JVM rather than different machine code for each type of machine. JVM executes the byte code generated by compiler and produce output. JVM is the one that makes java platform independent.	3	3	3
II.2	Constructor is a special method that gets invoked "automatically" at the time of object creation. Constructor has the same name as the class name. Constructor is normally used for initializing objects with default values unless different values are supplied.	3	3	3
II.3	1. Using packagename.* 2. Using packagename.classname 3. Using fully qualified name	3 X 1	3	3
II.4	If we make any class as final, we cannot extend it. Eg. final public class MyFinalClass { // class implementation }	2 + 1	3	3

QNo	Scoring Indicators	Split score	SubTotal	Total score
II. 5	1. Swing is a part of JFC (Java Foundation Classes) 2. Swing is built on top of AWT (Abstract Window Toolkit) 3. Swing components are lightweight 4. Swing components have pluggable look and feel 5. Swing components are platform independent 6. Swing components are included in the package javax.swing (Any 3)	3 X 1	3	3
II. 6	A listener is an object that is notified when an event occurs. For an event listener to work it must have been registered with one or more sources to receive notification and an appropriate method should have been implemented to process the event.	3	3	3
II. 7	1. add(Component c) - inserts a component on a container. 2. setSize(int width,int height) - sets the size (width and height) of the component. 3. setLayout(LayoutManager m) - defines the layout manager for the component. 4. setVisible(boolean status) - changes the visibility of the component, by default false. 5. setTitle(String text) -sets the title for a component (Any three) .	1+1+1	3	3
II. 8	1. By creating the object of JFrame class 2. By extending JFrame class.	1.5 x 2	3	3
II. 9	1. INTEGER/INT 2. TINYINT 3. SMALLINT 4. DECIMAL 5. CHAR 6. VARCHAR 7. TEXT 8. DATE: 9. TIME: 10. DATETIME 11. BOOLEAN	Any 6X1/2	3	3
II. 10	In JDBC, a ResultSet is an interface that represents the result of a database query. It provides methods to traverse and manipulate the data returned by the query. The ResultSet object is created when the executeQuery() method of the Statement object is called with an SQL query.	3	3	3
III PART C				42

QNo	Scoring Indicators	Split score	SubTotal	Total score
III	<p>1. Object 2. Class 3. Encapsulation 4. Abstraction 5. Polymorphism 6. Inheritance 7. Dynamic Binding 8. Message Passing</p> <p>Object Objects are real-world entities that have their own properties and behavior. It has physical existence. Eg: person, banks, company, customers etc</p> <p>Class A class is a blueprint or prototype from which objects are created. A class is a generalized description of an object.</p> <p>Encapsulation The wrapping up of data(variables) and function (methods) into a single unit (called class) is known as encapsulation.</p> <p>Abstraction Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.</p> <p>Polymorphism Polymorphism is the ability of a message to be displayed in more than one form.</p> <p>Inheritance The capability of a class to derive properties and characteristics from another class is called Inheritance.</p> <p>Dynamic Binding Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at runtime</p> <p>Message Passing A message for an object is a request for execution of a procedure. Message passing involves specifying the name of the object, the name of the method(message), and the information to be sent</p>	Listin g-1 Any 4 X1.5	1+6	7

QNo	Scoring Indicators	Split score	SubTotal	Total score
IV	<pre> import java.util.Scanner; class Person { String name; int age; public Person(String name, int age) { this.name = name; this.age = age; } public void printDetails() { System.out.println("Name: " + name); System.out.println("Age: " + age); } } public class Main { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Enter your name: "); String name = scanner.nextLine(); System.out.print("Enter your age: "); int age = scanner.nextInt(); Person person = new Person(name, age); person.printDetails(); } } </pre> <p>class Declaration – 1, Input - 2 Logic – 3 Output – 1</p>	7	7	7
V	<p>An exception is a condition that is caused by a runtime error in the program. When the JVM encounters an error such as divide by zero, it creates an exception object and throws it – as a notification that an error has occurred. (2 marks)</p> <p>Eg</p> <pre> class exceptionhandling { public static void main(String args[]) { int a[]={5,10}; int b=5; try { int x=a[2]/b-a[1]; } } } </pre>			

QNo	Scoring Indicators	Split score	SubTotal	Total score
	<pre>catch(ArithmeticException e){ System.out.println(" Division by zero"); } catch(ArrayStoreException e){ System.out.println(" Wrong data type"); } int y=a[1]/a[0]; System.out.println(" y=" +y); } }(5 marks)</pre>	2+5	7	7
VI	<p>Java supports definition of global methods and variables that can be accessed without creating objects of a class. Such members are called Static members</p> <p>They are prefixed with the keyword static</p> <p>Static variables: One copy per class – All objects share the same variable .</p> <p>E.g. static int count; (3 marks)</p> <p>Static methods: A class can have methods that are defined as static (e.g., main method). Static methods can be accessed without using objects. Also, there is NO need to create objects. They are prefixed with keyword “static”. (4 marks)</p>	3+4	3	
VII	<p>Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object.(2 marks)</p> <p>Different forms of inheritance:</p> <ol style="list-style-type: none"> 1.Single inheritance (only one super class) 2.Multiple inheritance (several super classes) 3.Hierarchical inheritance (one super class, many sub classes) 4.Multi-Level inheritance (derived from a derived class) 5.Hybrid inheritance (more than two types) (5 marks) 	2+1 X5	7	7
VIII	<p>The super keyword in Java is a reference variable which is used to refer immediate parent class object.</p> <p>Usage of Java super Keyword</p> <ol style="list-style-type: none"> 1. super can be used to refer immediate parent class instance variable. 2. super can be used to invoke immediate parent class method. 3. super() can be used to invoke immediate parent class constructor. 4. We can use super keyword to access the data member of parent class. It is used if parent class and child class have same attribute.(4 marks) 			

QNo	Scoring Indicators	Split score	SubTotal	Total score
	<p>Eg</p> <pre> class Person { private String name; public Person(String name) { this.name = name; } public void introduce() { System.out.println("Hi, my name is " + name); } } class Student extends Person { private int gradeLevel; public Student(String name, int gradeLevel) { super(name); this.gradeLevel = gradeLevel; } public void study() { System.out.println("I'm studying in grade " + gradeLevel); } } class MainC { public static void main(String[] args) { Student student = new Student("John", 9); student.introduce(); student.study(); } } </pre> <p>(3 marks)</p>	4+3	7	7
IX	<p>Method overloading and method overriding are two important concepts in Java's object-oriented programming paradigm. They are used to achieve polymorphism, which means the ability of an object to take many forms.</p> <p>Method Overloading: Method overloading allows multiple methods in a class to have the same name but with different parameters. It is achieved by changing the number of parameters or the data types of the parameters in the method signature.</p>	3.5 X 2	7	7
	<p>When a method is called, the compiler determines which method to execute based on the number of arguments and their data types. Method overloading is also known as compile-time polymorphism or static polymorphism.</p> <p>Method Overriding: Method overriding is a technique that allows a subclass to provide a specific implementation of a method that is already provided by its super class. It is achieved by creating a method in the subclass with the same name, return type, and parameters as the method in the super class. The method in the subclass must also have the same or a more accessible access level than the method in the super class. Method overriding is also known as runtime polymorphism or dynamic polymorphism.</p>			

QNo	Scoring Indicators	Split score	SubTotal	Total score
	<pre>import java.util.Scanner; interface Shape { void draw(); } class Rectangle implements Shape { private String name; public Rectangle(String name) { this.name = name; } public void draw() { System.out.println("Drawing rectangle"); } } public class Example { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Enter the name of the rectangle: "); String name = scanner.nextLine(); Rectangle rectangle = new Rectangle(name); rectangle.draw(); System.out.println("Name of the rectangle is: " + name); } } class Declaration – 2, interface - 2 implementation – 3</pre>	1 +3+3	7	7
XI	<p>1. Import the necessary Swing packages: To use Swing in a Java program, you need to import the necessary packages, such as javax.swing and java.awt.</p> <p>2. Create a JFrame object: The JFrame class represents the main window of a Swing application. You can create a JFrame object using the constructor or by extending the JFrame class.</p> <p>3. Add components to the JFrame: Once you have a JFrame object, you can add components to it, such as buttons, labels, text fields, and other controls.</p> <p>4. Set layout manager: Choose a layout manager that best suits the design of the interface.</p> <p>5. Set the properties of the JFrame: You can set various properties of the JFrame, such as its size, title, and visibility.</p> <p>6. Write event handlers: You can write event handlers to handle user input, such as button clicks or menu selections.</p>	1+2+1 +1+1+ 1	7	7

QNo	Scoring Indicators	Split score	SubTotal	Total score
XII	<pre>import javax.swing.*; import java.awt.*; import java.awt.event.*; public class ButtonExample extends JFrame implements ActionListener { private JButton button; public ButtonExample(){ setSize(200,200); setTitle("Button Example"); button = new JButton("Click Me"); add(button); button.addActionListener(this); setVisible(true); } public void actionPerformed(ActionEvent e) { if (e.getSource() == button) { JOptionPane.showMessageDialog(this, "Button clicked!"); } } public static void main(String[] args) { ButtonExample aob=new ButtonExample(); } } class Declaration – 1, creation of swing components - 3 Methods implementation – 3</pre>	7	7	7
XIII	<p>1.CREATE - This command is used to create a table.</p> <p>Syntax: CREATE TABLE table_name (column1 datatype, column2 datatype, column3 datatype,);</p> <p>Example: CREATE TABLE EMPLOYEE (Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE)(3 marks)</p> <p>2. ALTER - This command is used to modify the structure of an existing table. We can</p> <p>a) Add a new column to the table Syntax: ALTER TABLE table_name ADD column_name datatype; Example: ALTER TABLE Customers ADD Email varchar(255);</p> <p>b) Delete a column Syntax: ALTER TABLE table_name DROP COLUMN column_name; Example: ALTER TABLE Customers DROP COLUMN Email;</p>			

QNo	Scoring Indicators	Split score	SubTotal	Total score
XIII	<p>c) Rename a column Syntax: ALTER TABLE table_name RENAME COLUMN old_name to new_name; Example: ALTER TABLE Customers RENAME COLUMN Email to Mail</p> <p>d) Modify datatype of an attribute Syntax: ALTER TABLE table_name MODIFY COLUMN column_name datatype; Example: ALTER TABLE Customers MODIFY COLUMN Email varchar (50); (3 marks)</p> <p>3. DROP - It is used to delete both the structure and record stored in the table. Syntax: DROP TABLE table_name; Example: DROP TABLE Employee;(1 mark)</p>	3+3+1	7	7
XIV	<p>1. Load the JDBC driver: Before connecting to a database we need to load the appropriate JDBC driver for the database. We load the driver class by calling Class.forName() with the driver class name as an argument. For connecting to MySQL database, we write Class.forName("com.mysql.jdbc.Driver");</p> <p>2. Establish a connection: The JDBC DriverManager class defines objects which can connect Java applications to a JDBC driver. Its getConnection() method is used to establish a connection to a database. It uses a username, password, and a JDBC URL to establish a connection to the database and returns a connection object. Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/mydatabase", "username", "password");</p> <p>3. Create a statement: After establishing a connection, you can create a Statement object using the Connection.createStatement() method. The Statement object is used to execute SQL statements. Statement stmt = conn.createStatement();</p> <p>4. Execute SQL statements: You can execute SQL statements using the Statement.executeQuery() method. The executeQuery() method takes a sql query string as an argument and returns the result as a ResultSet object. ResultSet rs = stmt.executeQuery("SELECT * FROM mytable"); The executeUpdate() method executes the SQL INSERT,DELETE,UPDATE statements</p>			

QNo	Scoring Indicators	Split score	SubTotal	Total score
	<p>5. Process the results: We can retrieve the data from the ResultSet using the getXXX() methods, where XXX is the data type of the column. <code>while (rs.next()) { int id = rs.getInt("id"); String name = rs.getString("name"); System.out.println("ID: " + id + ", Name: " + name); }</code></p> <p>6. Close the connection: After all database operations are finished, the connection can be closed using the <code>Connection.close()</code> method. <code>conn.close();</code></p>	<p>1+2+1 +1+1+ 1</p>	<p>7</p>	<p>7</p>