S7

Apr-25

Scoring Indicators

7-18

COURSE NAME : OBJECT ORIENTED PROGRAMMING

COURSE CODE : 4131 **QID : 2103230199**

Qn. No.	Scoring Indicators	Split score	Total Score
	PART A		**9**
1	Class	1	1
2	Java is platform independent because java class files can run on all operating systems with the help of JVM without any modifications	1	1
3	extends	1	1
4	Multiple inheritance	1 mark	1
5	Jlabel, JtextField, Jbutton, Jlist, Jcheckbox, Jcombobox, Jpanel, Jtextarea	Any two 1 mark	1
6	Change in the state of an object is called event	1 mark	1
7	JButton	1 mark	1
8	Java Database Connectivity	1 mark	1
9	forName() method	1 mark	1
	PART B		**24**

1	class BankAccount{
	int accno;
	String name;
	BankAccountt(int r, String n) {
	accno = r;
	name = n;
	}}
		3 marks	3

2	Public:anything public is accessible to anywhere Private: anything private is only accessible in the class they are declared Protected: anything protected is accessible outside the package but only to child classes and default is accessible only inside the package.	3 points 3 marks	3

3	class A
	{
	}
	class B extends A
	{
	}
	class C extends B
	{
	}
	class D extends B
	{
	}
		3 marks	3

4	final is a non-access modifier applicable only to a variable, a method or a class • The contents of final variables cannot be modified.	3 points marks	

<table>
<tr><td></td><td>

When a method is declared as final then it cannot be overridden by subclasses
When a class is declared as final then it cannot be subclassed i.e. no any other class can extend it

</td><td></td><td>3</td></tr>
<tr><td>5</td><td>lightweight - as swing doesn't use native resources, it is light-weighted.

platform-independent - it doesn't depend on any platform thus, the swing is platform-independent.

customizable - it can be designed and made according to the requirement of the programmer.

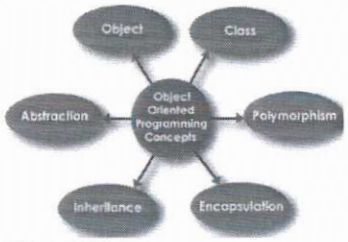
configurable - as we see that the applications that have regular updates with new features swing also provides a facility to configure existing applications with variable features.</td><td>Any 3 points 3 marks</td><td>3</td></tr>
<tr><td>6</td><td>
<table>
<tr><td>TextEvent</td><td>TextListener</td></tr>
<tr><td>WindowEvent</td><td>WindowListener</td></tr>
<tr><td>ActionEvent</td><td>ActionListener</td></tr>
</table>
</td><td>3 points 3 marks</td><td>3</td></tr>
<tr><td>7</td><td>Mouse events occur with mouse movements in forms and controls. Following are the various mouse events related with a Control class –

MouseDown – it occurs when a mouse button is pressed
MouseEnter – it occurs when the mouse pointer enters the control
MouseHover – it occurs when the mouse pointer hovers over the control
MouseLeave – it occurs when the mouse pointer leaves the control
MouseMove – it occurs when the mouse pointer moves over the control
MouseUp – it occurs when the mouse pointer is over the control and the mouse button is released
MouseWheel – it occurs when the mouse wheel moves and the control has focus

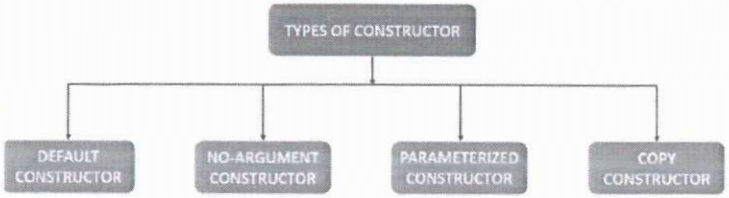
</td><td>Any 3 mouse events 1.5 marks

Any three keyboard events 1.5 marks</td><td>3</td></tr>
</table>

	Following are the various keyboard events related with a Control class – • KeyDown – occurs when a key is pressed down and the control has focus • KeyPress – occurs when a key is pressed and the control has focus • KeyUp – occurs when a key is released while the control has focus		
8	Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information. Example Microsoft SQL Server, Oracle Database, MySQL	Defention 2marks marks,example 1 marks	3
9	1. DDL – Data Definition Language List of DDL commands: • CREATE: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers). • DROP: This command is used to delete objects from the database. • ALTER: This is used to alter the structure of the database. • TRUNCATE: This is used to remove all records from a table, including all spaces allocated for the records are removed. • COMMENT: This is used to add comments to the data dictionary. • RENAME: This is used to rename an object existing in the database. 2. DML – Data Manipulation Language List of DML commands: • INSERT : It is used to insert data into a table. • UPDATE: It is used to update existing data within a table. • DELETE : It is used to delete records from a database table. • LOCK: Table control concurrency. 3. DCL – Data Control Language List of DCL commands: • GRANT: This command gives users access privileges to the database. • REVOKE: This command withdraws the user's access privileges given by using the GRANT command.	3 points 1.5 marks each. 2 commands from 3 points.1.5 marks	3

10	Interface PreparedStatement. An object that represents a precompiled SQL statement. A SQL statement is precompiled and stored in a PreparedStatement object. This object can then be used to efficiently execute this statement multiple times. prepareStatement() method of Connection interface is used to return the object of PreparedStatement PreparedStatement stmt=con.prepareStatement("insert into Emp values(?,?)");	Defenitio n 1.5 marks Eg :1.5 marks	3
	PART C		**42**
III 1	 CLASS • A class is a blueprint or prototype from which objects are created • A class is a generalized description of an object. No physical existence. • Acts as a type or category. OBJECT • An object is an instance of a class • Objects are real-world entities that has their own state and behavior. • An object has physical existence ABSTRACTION Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. ENCAPSULATION The wrapping up of data(variables) and functions (methods) that operates on the data, into a single unit (called class) is known as encapsulation. • It is also called "information hiding", hides the internal representation, or state, of an object from the outside through access modifies (Private / Public). INHERITANCE The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is the process by which objects of one class acquired the properties of objects of another classes POLYMORPHISM The word polymorphism means having many forms. • Refers to a programming language's ability to process objects differently depending on their data type or class	Any 4 property 7 marks	7

	An operation may exhibit different behaviors in different instances.		

III 2	A constructor is similar to method and it is invoked at the time creating an object of the class, it is generally used to initialize the instance variables of a class. The constructors have same name as their class and, have no return type.
	 
	
	1. **Default Constructor**
	A default constructor is automatically created at the time of object creation if no constructor is explicitly declared in the class.
	Eg:
	class Rectangle{
	double length = 10;
	double breadth = 30;
	}
	/object creation creates the default constructor automatically without visible construct definition
	Rectangle myrec = new Rectangle();
	2. **No-argument Constructor**
	No-argument constructors are visible in the class, unlike default constructors. Also, it has **a constructor definition but without arguments** (or parameters).
	3. **Parameterized Constructor**
	Unlike, No-argument constructor, it contains parameters (or arguments) in the constructor definition and declaration. More than one argument can also be passed through parameterized constructor in Java.
	4. **Copy Constructor**
	A copy constructor is a special type constructor which creates an object using another object of the same class. In return, it gives a copy of an already created object that is copied.
		2x3
		1
			7

Example:

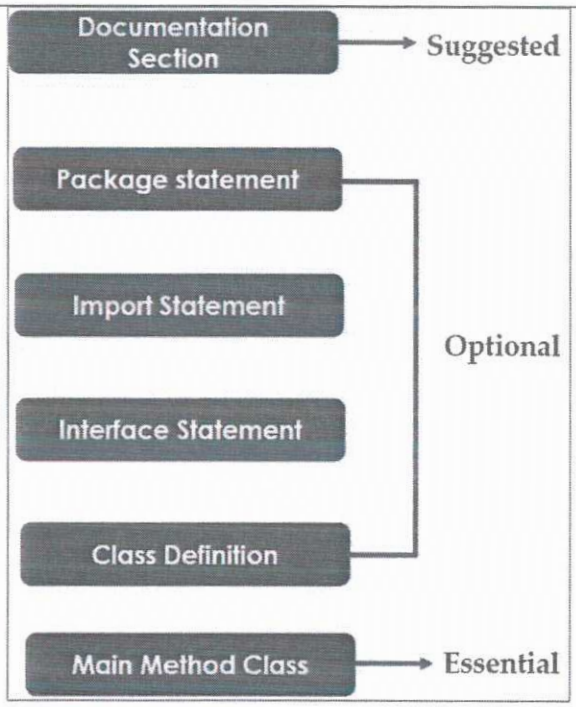
```
class Rectangle
{
   double length;
   double breadth;

   Rectangle()
   {
      length = 10;
      breadth = 20;
   }

   Rectangle(double l, double b)
   {
     length = l;
     breadth = b;
   }
Rectangle(Rectangle rect)
  {
   System.out.println("Copy Constructor Invoked!!");
   length = rect.length;
   breadth = rect.breadth;
  }
class Rectangle_Main{

  public static void main(String[] args)
  {
```

	Rectangle rec1=new Rectangle()
	Rectangle rec2 = new Rectangle(10,20);
	//passing the parameter to the copy constructor
	Rectangle rec3 = new Rectangle(rec1);
	}
	}		

III 3	
	/*****First Java Program: HelloWorld.java *****/
	//* Author: Mary
	"main" method
	public class HelloWorld
	{
	public static void main(String[ ] args)
	{
	System.out.println("Hello, world!");
	}
	}
	Document section (optional)
	• Add comments describing the details of author, purpose,
		7x1	7

	licence, etc
	
	Package Statement (optional)
	• You can create a package with any name to organize the classes of that file.
	• A package is to group classes that are defined by a name.
	• It is like a folder in which all the classes present in that file are kept.
	• It is declared as: **package package_name**
	Import Statement (optional)
	• This line indicates that if you want to use a class of another package, then you can do this by importing it directly into your program.
	• E.g. **import java.io.*;// importing all the classes from java.io package**
	Interface statement (optional)
	• Interfaces are like a class that includes a group of method declarations.
	• It's an optional section and can be used when programmers want to implement multiple inheritances within a program.
	Class Definition (optional)
	• A Java program file may contain several class definitions.
	• But only one public class is allowed in a file
	• File name and the public class name should be same.		

III 4	• Exception is an abnormal condition. • In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime. • Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc. • Suppose there are 10 statements in your program and there occurs an exception at statement 5, the rest of the code will	7x1	7

not be executed i.e. statement 6 to 10 will not be executed.

- If we perform exception handling, the rest of the statement will be executed. That is why we use exception handling in Java.

```
statement 1;
statement 2;
statement 3;
statement 4;
statement 5;//exception occurs
statement 6;
statement 7;
statement 8;
statement 9;
statement 10;
```

Keyword	Description
try	The "try" keyword is used to specify a block where we should place an exception code. It means we can't use try block alone. The try block must be followed by either catch or finally.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the necessary code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It specifies that there may occur an exception in the method. It doesn't throw an exception. It is always used with method signature.

```
public class JavaExceptionExample{
 public static void main(String args[]){
  try{
    //code that may raise exception
    int data=100/0;
  }catch(ArithmeticException e){System.out.println(e);}
  //rest code of the program
  System.out.println("rest of the code...");
  }
}
```

	In the above example, 100/0 raises an ArithmeticException which is handled by a try-catch block.		

III 5	class Person
	{
	String name;
	String address;
	Person(String n,String a)
	{
	name=n;
	address=a;
	}
	}
	class Student extends Person
	{
	int rollno;
	String clss;
	Student(String n,String a,int r,String c)
	{
	super(n,a);
	rollno=r;
	clss=c;
	}}
	class Student_main
	{
	public static void main(String s[])
	{
	Student s1=new Student("Raj","Delhi",101,"Computer Engineering");
	}}
		7 x 1	7

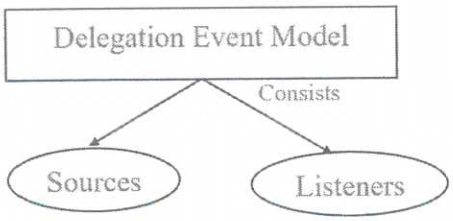
III 6	class BaseClass
	{
	public void myFunction()
	{
	}
	}
	class DerivedClass extends BaseClass
	{
	public void myFunction()
	{
	}}
	Here in both classes method with same signature is used.
	If the method is invoked with DerivedClass object, Then derived class version of myFunction is called. That is in derived class the myFunction of BaseClass is overriden.
	DerivedClass d1=new DerivedClass() ;
		7X1	7

	D1.myFunction() ;
	But if the method is invoked with BaseClass object reference pointing DerivedClass object, then compiler does not know at compile time which version of method is invoked.
	BaseClass b1 ;
	B1=new DerivedClass() ;
	B1.myFunction() ; // DerivedClass version of method is invoked.
	If
	B1=new BaseClass() ;
	B1.myFunction() ; // BaseClass version of method is invoked.
	At runtime only it is known that whether DerivedClass version or BaseClass version of myFunction is invoked depending on the type of object the reference is pointing. This is called dynamic method dispatch and is known as Runtime Polymorphism .		

III 7.	interface MotorBike
	{
	int speed=50;
	public void totalDistance();
	}
	interface Cycle
	{
	int distance=150;
	public void speed();
	}
	public class TwoWheeler implements MotorBike,Cycle
	{
	int totalDistance;
	int avgSpeed;
	public void totalDistance()
	{
	totalDistance=speed*distance;
	System.out.println("Total Distance Travelled : "+totalDistance);
	}
	public void speed()
	{
	int avgSpeed=totalDistance/speed;
	System.out.println("Average Speed maintained : "+avgSpeed);
	}
	public static void main(String args[])
	{
	TwoWheeler t1=new TwoWheeler();
	t1.totalDistance();
	t1.speed();
	}
		7X1	7

	}		

III 8	Book.java
	
	package pack1;
	public class Book
	{
	String name;
	String author;
	String publisher;
	public void read(String n,String a,String p)
	{
	name=n;
	author=a;
	publisher=p;
	}
	public void show()
	{
	System.out.println("Name"+name+"author"+author+"Publisher"+publisher);
	}
	}
	
	Get.java
	
	package pack2;
	import pack1.Book;
	public class
	{
	public static void main(String s[])
	{
	Book b1=new Book();
	B1.read();
	B1.show();
	}
	}
		7x1	7

III 9	Event Handling It is a mechanism to control the events and to decide what should happen after an event occur. To handle the events, Java follows the *Delegation Event model.* Delegation Event model • It has Sources and Listeners.  *Delegation Event Model* • Source: Events are generated from the source. There are various sources like buttons, checkboxes, list, menu-item, choice, scrollbar, text components, windows, etc., to generate events. • Listeners: Listeners are used for handling the events generated from the source. Each of these listeners represents interfaces that are responsible for handling events. To perform Event Handling, we need to register the source with the listener. Registering the Source With Listener Different Classes provide different registration methods. Syntax: addTypeListener() where Type represents the type of event. Flow of Event Handling 1. User Interaction with a component is required to generate an event. 2. The object of the respective event class is created automatically after event generation, and it holds all information of the event source. 3. The newly created object is passed to the methods of the registered listener. 4. The method executes and returns the result.	1 3 3	7

III 10	Event Classes	Listener Interfaces	Any 7 points 7 marks	7
	ActionEvent	ActionListener		
	MouseEvent	MouseListener and MouseMotionListener		
	MouseWheelEvent	MouseWheelListener		
	KeyEvent	KeyListener		
	ItemEvent	ItemListener		
	TextEvent	TextListener		
	AdjustmentEvent	AdjustmentListener		
	WindowEvent	WindowListener		
	ComponentEvent	ComponentListener		
	ContainerEvent	ContainerListener		
III 11	There are 5 steps to connect any java application with the database using JDBC. These steps are as follows: o Register the Driver class o Create connection o Create statement o Execute queries o Close connection 1) Register the driver class The **forName()** method of Class class is used to register the driver cl method is used to dynamically load the driver class. Syntax of forName() method **public static void** forName(String className)**throws** ClassNotFoun dException Example to register the OracleDriver class Here, Java program is loading oracle driver to esteblish database connection.			7

Class.forName("oracle.jdbc.driver.OracleDriver");

2) Create the connection object

5

The **getConnection()** method of DriverManager class is used to connection with the database.

Syntax of getConnection() method

1) **public static** Connection getConnection(String url)**throws** SQLException

2) **public static** Connection getConnection(String url,String name,String password)

throws SQLException

Example to establish connection with the Oracle database

```
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","password");
```

3) Create the Statement object

The createStatement() method of Connection interface is used t statement. The object of statement is responsible to execute queries database.

Syntax of createStatement() method

public Statement createStatement()**throws** SQLException

Example to create the statement object

```
Statement stmt=con.createStatement();
```

4) Execute the query

The executeQuery() method of Statement interface is used to execute to the database. This method returns the object of ResultSet that can b get all the records of a table.

Syntax of executeQuery() method

public ResultSet executeQuery(String sql)**throws** SQLException

Example to execute query

```
ResultSet rs=stmt.executeQuery("select * from emp");

while(rs.next()){
System.out.println(rs.getInt(1)+" "+rs.getString(2));
}
```

5) Close the connection object

By closing connection object statement and ResultSet will be automatically. The close() method of Connection interface is used to

	connection. Syntax of close() method **public void** close()**throws** SQLException Example to close connection con.close();		
III 12	CREATE It is used to create a new table in the database. Example: CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VA RCHAR2(100), DOB DATE); DROP: It is used to delete both the structure and record stored in the table. Syntax DROP TABLE table_name; Example DROP TABLE EMPLOYEE; ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute. Syntax: To add a new column in the table ALTER TABLE table_name ADD column_name COLUMN-definition; To modify existing column in the table: ALTER TABLE table_name MODIFY(column_definitions....); EXAMPLE ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20)); ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(2 0)); INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.	Any 7 commands 7 marks	7

Syntax:

INSERT INTO TABLE_NAME
(col1, col2, col3,.... col N)
VALUES (value1, value2, value3, ... valueN);

UPDATE: This command is used to update or modify the value of a column in the table.

Syntax:

UPDATE table_name SET [column_name1= value1,...column_name N = valueN] [WHERE CONDITION]

For example:

UPDATE students
SET User_Name = 'Sonoo'
WHERE Student_Id = '3'

DELETE: It is used to remove one or more row from a table.

Syntax:

DELETE FROM table_name [WHERE condition];

For example:

DELETE FROM javatpoint
WHERE Author="Sonoo";

SELECT: This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

Syntax:

SELECT expressions
FROM TABLES
WHERE conditions;