**Examination Answer Book**

**UNIVERSITY EXAMS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| REGISTRATION NUMBER | | | | | | | | | VU-BIT-2407-0862-DAY | | | | | | |
| Title of The Program (eg BBA, BSC, BPH, BSWA) | | | | | | | | | | | | | BIT | | |
| Bachelor of Information Technology | | | | | | | | | | | | | | | |
| Department | | | | Other Depts in Faculty of Science and Technology | | | | | | | | | | | |
| Faculty | Faculty of Science and Technology | | | | | | | | | | | | | | |
| Year Of study (YrI , YrII, YrIII, or YrIV) | | | | | | | | | | | 1 | | | | |
| Module Code and Name | | | | | | | 1301 ST | | | | | | | | |
| Object Oriented Programming | | | | | | | | | | | | | | | |
| Semester | | | 3 | | | | | | | | | | | | |
| (Copy from the heading to the Examination Paper) | | | | | | | | | | | | | | | |
| Retake: | | Yes | | |  | | | No | |  | | (Tick whichever is applicable) | | | |
| Date of examination | | | | | | Sat Jul 05 2025 09:00:00 GMT+0300 (East Africa Time) | | | | | | | | | |
|  | | | | | |  | | | | | | | | | |
| **DIRECTIONS TO CANDIDATES (Turn to page ii for more instructions).** | | | | | | | | | | | | | **FOR USE BY EXAMINERS ONLY** | | |
| **Question Number** | **Internal Examiner** | **External Examiner** |
| 1. Leave margin blank. 2. Begin each answer on a fresh page. 3. Write the number of each question and theCandidate's Number at the top of each page. 4. Write the numbers of the questionswhich you have attempted, with subsections where necessary, in the spacesprovided below | | | | | | | | | | | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **TOTAL** |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NUMBER OF QUESTIONS** you have answered in the order in which you have written them | | | | | | | | |
| 1 | 2 | 3 | 5 |  |  |  |  |  |

**How and where should I submit my examination script?**

Every student will be required to attend their examination via [VClass Students Portal](https://vclass.ac/) E.g. you go to [www.vclass.ac](http://www.vclass.ac) and login, to your account, then on the left sidebar menu **click on Examinations**.

Under examinations you will see the following: -

1. Instructions for that particular examination with time required to finish your examination as per instructions,
2. A student will be required to download the question paper and the answer sheet provided by the university within the same module examination, or a student can be required to attempt structured questions within the system depending on how the examination was set.
3. Submission of answered questions is done,
4. Student is required to click to **consent** to show that the answered exam belongs to them.
5. **Note** that if an examination is for download, a student will be required to download the question paper and answer sheet, write their examination within the given stipulated time.
6. Required to scan and upload back the answered booklet through the same portal as per format available.
7. Examinations uploaded will directly be received by the Registry department.
8. Students here are required to use [VClass e-Learning system](https://vclass.ac)for all examinations and for any failure they can contact the Registry department for guidance.
9. No late submission will be accepted.

**Avoid any examination malpractice because this will attract severe penalties such as invalidating the exams answered script whose consequences will attract retakes.**

NUMBER (2)

String Immutability in Java

strings are immutable in Java meaning once a String object is created the value cannot be changed.

2b

String title = "";

Or

String author = new String();

c)

When patrons search for books, they might enter queries in different cases ie "uganda", "UGANDA", "Uganda".

if (bookTitle.equalsIgnoreCase(userInput)) {} the case wont matter its will still work

Use of toLowerCase():

if (bookTitle.toLowerCase().contains(userInput.toLowerCase())) {}

d) Object Comparison Example

String a = "Library";

String b = new String("Library");

System.out.println(a == b);

System.out.println(a.equals(b));

Explanation:

a == b compares references and returns false because b points to a new object.

a.equals(b) compares values and returns true as both strings have the same content.

Number (5)

a) Five Core OOP Principles and Application in URA’s EFRIS System

Encapsulation where we wrap data and methods into a single unit restricting direct access.

Application: EFRIS encapsulates sensitive data like TINs and invoice amounts within secure classes to prevent unauthorized access or modification.

Abstraction

Definition: Hiding complex implementation details and exposing only essential features.

Application: The EFRIS user interface abstracts away the complexities of tax computation, exposing only relevant input/output fields to users.

Inheritance

Definition: A class can inherit properties and behaviors from another class.

Application: A general Tax Category class could be inherited by Retailer, Wholesaler, etc., to share VAT computation behavior and override as needed.

Polymorphism

Definition: The ability to present the same interface for different underlying forms (data types).

Application: EFRIS applies different VAT calculation logic using overridden methods in subclasses based on taxpayer type.

reporting that can be developed and maintained independently.

b) Encapsulation in EFRIS using a Java Class

Encapsulation is used to hide internal transaction data from direct access.

public class TransactionRecord {

private String buyerTIN;

private String sellerTIN;

private double invoiceAmount;

private String timestamp;

public TransactionRecord(String buyerTIN, String sellerTIN, double invoiceAmount, String timestamp) {

this.buyerTIN = buyerTIN;

this.sellerTIN = sellerTIN;

this.invoiceAmount = invoiceAmount;

this.timestamp = timestamp;

}

public String getBuyerTIN() {

return buyerTIN;

}

public void setBuyerTIN(String buyerTIN) {

this.buyerTIN = buyerTIN;

}

public String getSellerTIN() {

return sellerTIN;

}

public void setSellerTIN(String sellerTIN) {

this.sellerTIN = sellerTIN;

}

public double getInvoiceAmount() {

return invoiceAmount;

}

public void setInvoiceAmount(double invoiceAmount) {

this.invoiceAmount = invoiceAmount;

}

public String getTimestamp() {

return timestamp;

}

public void setTimestamp(String timestamp) {

this.timestamp = timestamp;

}

}

This structure ensures that sensitive fields are only accessible via public methods, enforcing control and validation if necessary.

c) Polymorphism for Different VAT Calculation

Polymorphism enables the EFRIS system to process different types of taxpayers while calling the same method (calculateVAT()), with behavior varying depending on the taxpayer category.

Number (3)

a)

Exception:

An exception in Java is an event that disrupts the normal flow of a program’s execution. It usually occurs due to logical errors or unexpected events (e.g., division by zero, file not found, invalid input).

Exception Handling:

Java provides a mechanism to handle such runtime errors using try, catch, finally, and throw. This allows the program to continue running rather than crash.

Importance in AirQO System:

In the AirQO dashboard, exception handling ensures that errors like sensor data misreads, division by zero, or null values do not crash the entire monitoring system. Instead, they are logged and handled gracefully.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b) Division by Zero Exception in AQI Calculation

Exception Thrown: ArithmeticException

Why it occurs:

In Java, dividing a number by zero (especially integers) triggers an ArithmeticException at runtime.

For example:

java

CopyEdit

int averageAQI = totalAQI / 0;

Explanation in AirQO Context:

If the number of readings is zero during AQI averaging, dividing by it without validation will throw an exception, disrupting the air quality calculation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String Comparison Output Analysis

String author1 = "sidney";

String author2 = "sidney";

String author3 = new String("sidney");

System.out.println(author1 == author3

System.out.println(author1.equalsIgnoreCase(author2));

• author1 == author3 → false because == checks for reference equality (different memory objects).

• author1.equalsIgnoreCase(author2) → true because equalsIgnoreCase() ignores case during content comparison.

c)

Using while loop:

java

CopyEdit

int day = 1;

while (day <= 30) {

System.out.println("Tracking PM2.5 - Day " + day);

day++;

}

Using for loop:

java

CopyEdit

for (int day = 1; day <= 30; day++) {

System.out.println("Tracking PM2.5 - Day " + day);

}

d) Java Program: Generate AQI Readings, Find Median, Count Hazardous Days

import java.util.\*;

public class AirQODashboard {

public static void main(String[] args) {

int[] aqiReadings = new int[30];

Random random = new Random();

for (int i = 0; i < 30; i++) {

aqiReadings[i] = random.nextInt(300) + 1; // 1 to 300

}

System.out.println("AQI Readings for 30 days:");

System.out.println(Arrays.toString(aqiReadings));

Arrays.sort(aqiReadings);

double median;

if (aqiReadings.length % 2 == 0) {

median = (aqiReadings[14] + aqiReadings[15]) / 2.0;

} else {

median = aqiReadings[aqiReadings.length / 2];

}

System.out.println("Median AQI: " + median);

int hazardousCount = 0;

for (int aqi : aqiReadings) {

if (aqi > 200) {

hazardousCount++;

}

}

System.out.println("Number of hazardous days (AQI > 200): " + hazardousCount);

}

}