VIT-AP UNIVERSITY

Apply Knowledge, Improve Life!

QUESTION PAPER

Name of the Examination: WINTER 2022-2023 – FAT

Course Code: CSE2005

Course Title: Object Oriented Programming

Set number: 12

Date of Exam: 16/06/2023 (AN) (A2

Duration: 120 Minutes

Total Marks: 60 (5*12=60)

Q1. Write a java program for Banking System. Apply the inheritance concept to represent various types of bank accounts, such as savings accounts, checking accounts, and credit card accounts, with shared functionalities like deposit, withdrawal, and balance inquiry.

Create a base class Bankaccount with customer details like name, cust_id, account_number and balance. Create a Savings_account class inheriting from BankAccount class. Calculate the Interest in this class. Create a Checking_account class inheriting from BankAccount. Check the overdraft limit in this class. Create a Credit_card_account class inheriting from BankAccount. Check the credit card limit and calculate the interest in this class. Finally display all the details. **(12M)**

Q2. Write a Java program for patient database. How can a HashMap be employed to represent a patient database, where each patient is identified by their unique ID, and associated with their name, age, gender and other relevant information?

Add 10 patients' details as an element to the HashMap and display it. Change any 2 patients' details and display the HashMap. Remove any four patients' details and display the HashMap. **(12M)**

Q3. Write a Java program for developing a multi-threaded application that simulates a task scheduler. The application creates five threads (t1, t2, t3, t4, and t5), each representing a scheduled task. Each thread is assigned a priority and performs a specific operation. The application must handle interrupts and raise exceptions for specific scenarios.

Write a Java program that accomplishes the following: Create five threads (t1, t2, t3, t4, and t5) and set their priorities as follows: t1 with priority 9, t2 with priority 6, t3 with priority 7, t4 with priority 5, and t5 with priority 8. Implement a custom Task class that extends the Thread class. Each task should have a unique identifier and a specific operation to perform. In the run method of each task, simulate some work (assume some task to be performed) by sleeping the thread for a specific duration. Handle interrupts by catching the InterruptedException and printing an appropriate message. In the main method, start all the threads. After starting the threads, interrupt t4 from t5. Handle the interrupt by catching the InterruptedException and raising an exception with the message "Interrupted by t5". Display the thread name, priority, and state for each thread after the interrupt. Use the getName, getPriority, and getState methods to obtain this information. Handle any exceptions that may occur during the execution of the program and print the exception message. **(12M)**

Q4. Write a Java program for developing a ticket booking system for a cinema, where multiple customers can book seats simultaneously. The system must ensure that seats are allocated accurately and prevent double bookings. Implement inter-thread communication techniques to handle seat booking requests and ensure synchronization between customer threads (Make necessary assumptions). **(12M)**

Q5. Write a program to collect and display the employee details (Emp_Name, Emp_Id., Gender, Date_of _Birth, Place_of_Birth, Age, and Salary) using JavaFX. In Salary calculation collect the baseSalary, allowancePercentage is 0.15, deductionPercentage is 0.05, and the bonus is 5000. Display the totalSalary. Use at least four different controls while the GUI is designed using JavaFX. **(12M)**

Formula:

allowance = baseSalary * allowancePercentage

deduction = baseSalary * deductionPercentage

totalSalary = baseSalary + allowance - deduction + bonus

QP MAPPING

Q. No.	Module Number	CO Mapped	PO Mapped	PEO Mapped	PSO Mapped	Marks
Q1	1	1	1	4	1	12
Q2	4	4	2,5	2	1	12
Q3	5	5	3	3	1	12
Q4	5	5	3	3	1	12
Q5	6	6	2,3,5	3	1	12

VIT-AP UNIVERSITY

Apply Knowledge, Improve Life!

QUESTION PAPER

Name of the Examination: WINTER 2022-2023(Freshers)–FAT

Course Code: CSE2005 **Course Title: Object-Oriented Programming**

Set number: 13 **Date of Exam: 17/06/2023(FN)**

Duration:120 min **Total Marks: 60** (B1)

Instructions:

1. Assume data wherever necessary.
2. Any assumptions made should be clearly stated.

Q1. Create an interface called Advertising_Agency that holds the methods like a abstract methods like booking() and billing ().Create two classes called Company_A and Company_B that implement the interface. The abstract methods of the interface are implemented in the class for like booking() ←that inputs the details of the advertisement requirements and billing()←that computes the total amount for advertisement. Each company feeds their requirement as **(12 marks)**

Company A:	**Company B:**
mode of advertisement: TV	mode of advertisement: Radio
Time of advertisement: 2 min	Time of advertisement: 1 min
Quality of the advertisement: High	Quality of the advertisement: High

Compute the bill based on the following criteria
Tv: Rs.10000/-
Radio: Rs 5000/-
Time: Rs. 3000/- per min (both Tv and Radio)
Quality : high: Rs.1000/-; Low : Rs.500/-

Q2. A departmental store maintains the details of a few products including the product id, product name, packed date, quantity_available and rate_of_the_product. **(12 marks)**

- Create a hash set for the list of products and find the availability of a particular product based on the name of the product.
- Write a method that identifies the product which has a minimum quantity of 5 or less.

VIT-AP UNIVERSITY

Apply Knowledge, Improve Life!

QUESTION PAPER

Name of the Examination: WIN 2022-2023 – FAT

Course Code: CSE2005
Set number: 2
Duration: 120 mins

Course Title: Object Oriented Programming
Date of Exam: 14/06/2023 (FN)(B2)
Total Marks: 60

Instructions:

1. Assume data wherever necessary.
2. Any assumptions made should be clearly stated.

Q1. Create a class **Prime** with one data member called **num** of type int and a static method **isPrime()** to return true if data member is prime otherwise it should return false. Create another class **PrimeR** from **Prime** which overrides method **isPrime()** and this override method uses recursion to finds and returns whether super class data member is prime(true) or not(false). Create a test class to test the entire hierarchy and show the power of dynamic method dispatch. **(12M)**

Q2. Create a generic interface **Result** where Type are represented as **Regno** and **Mark** with abstract method **result (Regno,Mark).** Create a generic class **OOPs** that implements **result (Regno,Mark)** which find out the score of Multiple Choice Questions (mcq) question. Create ArrayList to store the answer keys of 10 mcq. Take student roll no, question no and its option of the mcq as user input and find the student score and percentage. Write a java program for the above scenario. Import appropriate packages for the program. **(12M)**

Sample Input/Output

Enter your roll number: 22bce8245
Enter your answer for question 1: b
Enter your answer for question 2: b
.
.
Enter your answer for question 10: b
Your score: 3
Percentage: 30.0%

Q3. Write java program that runs three threads T1,T2, and T3 simultaneously. class **MaxThread** is to find the maximum element, class **MidThread** is to find the mid element, and class **OddMinThread** it to find the odd minimum element from the same array. It includes the custom exception **"MidValueNotFoundException"** for finding mid element when the length of the array is even. Create **MainThread** class, that create instances of the three threads, start them, and then ensure that the main thread waits for the completion of each thread. Finally, after all the threads have finished, it will print a message indicating that the **"Main thread has finished"**. **(12M)**

Sample Input/Output

Arr= [1, 9, 3, 4, 5, 8, 7, 6, 2]
Mid element: 5
Odd minimum element: 1
Maximum element: 9
Main thread finished

Arr= [10, 9, 3, 4, 5, 8, 7, 6, 2,1]
Mid value not found: Array length is even
Maximum element: 10
Odd minimum element: 1
Main thread finished

Q4. Write a multi-thread java program using synchronization to find the power of the elements of an array. Create a **Arrpower** class that use synchronization block for finding the power of the numbers. Define a method **void findPowers(int[])** to display the power of the elements. Create a user thread **T1** to find the powers of even element and create another user thread **T2** to find powers of the odd elements of the array. Ensures that only one thread can access the synchronized block at a time. Create T1 and T2 by using **anonymous inner classes.** **(12M)**

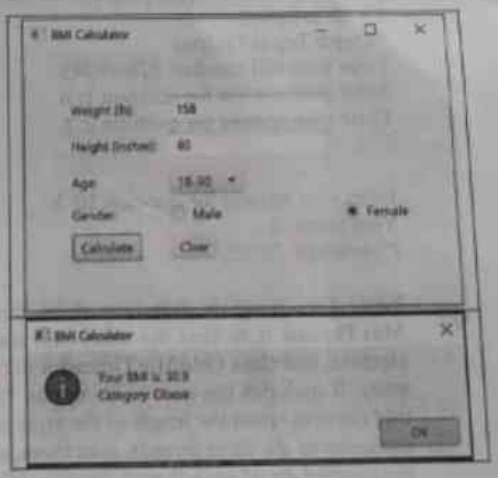
Sample Input/Output

Arr=[1,2,3,4,5]

T1: Powers of even elements	T2: Powers of odd elements
4	1
16	9
	25

Q5. Write a program to develop JavaFX application to add the components as shown in the below figure. Upon clicking on calculate button, an action to be performed to calculate BMI of a person. Also your application should display whether person is with underweight or over weight based on calculated BMI value, using alert box. Besides, upon clicking on Clear button, your application should reset all the text filed present in the scene. Consider age and gender for the BMI calculation. The age ranges are in between 2-17, 18-30, 31-50 and 51-100 years. The formula for BMI calculation, range, and category is provided below. Hints: US units : **BMI=(weight(lb)/height^2)*703(inches)** **(12M)**

BMI range	Category
16.0 - 18.4	Underweight
18.5 - 24.9	Normal
25.0 - 29.9	Overweight
>= 30	Obese



QP MAPPING

Q. No.	Module Number	CO Mapped	PO Mapped	PEO Mapped	PSO Mapped	Marks
Q1	1,2,3	1,2,3	1,2,11	2,4	-	12
Q2	4	1,2,3	1,2,11	2,4	-	12
Q3	3,5	4,5	2,3,5	2,3	-	12
Q4	5	5	3	3	-	12
Q5	6	6	2,3,5	3	-	12

VIT-AP UNIVERSITY
Apply Knowledge, Improve Life!

QUESTION PAPER

Name of the Examination: WINTER 2022-2023 – FAT

Course Code: CSE2005 **Course Title: Object Oriented Programming**

Date of Exam: A **Set Number:** 19/06/2023 (FN) (4)

Duration: 120 min **Total Marks: 60**

Instructions:

1. Assume data wherever necessary.
2. Any assumptions made should be clearly stated.

Q1. Create an interface called Playable with the following methods: void play(): Prints "Playing" to the console. void stop(): Prints "Stopped" to the console. Create two classes, AudioPlayer and VideoPlayer, that implement the Playable interface. The AudioPlayer class should implement the play() and stop() methods to handle audio-specific operations, such as playing and stopping audio playback. The VideoPlayer class should implement the play() and stop() methods to handle video-specific operations, such as playing and stopping video playback.In the main class, create instances of the AudioPlayer and VideoPlayer classes. Call the play() and stop() methods for each player to demonstrate their functionality. **(12M)**

Sample Input: No specific user input is required for this program.

Sample Output:

Playing audio
Stopped audio
Playing video
Stopped video

Q2. Create a program that keeps track of a student database using a HashMap. The program should allow the user to perform the following actions using switch case **(12M)**

- ✓ Add a student to the database by entering their ID (integer) and name (string).
- ✓ Retrieve and display the name of a student based on their ID.
- ✓ Remove a student from the database based on their ID.
- ✓ The program should continue running until the user chooses to exit

Sample Input/output:

Student Database
---
1. Add student
2. Retrieve student
3. Remove student
4. Exit

Enter your choice: 1
Enter student ID: 123
Enter student name: John Doe
Student added successfully.

VIT-AP UNIVERSITY

Apply Knowledge. Improve Life!

QUESTION PAPER

Name of the Examination: WINTER 2022-2023 – FAT

Course Code: CSE2005 **Course Title:** Object Oriented Programming

Set number: 5 **Date of Exam:** 19/06/2023 (AN)

Duration: 120 minutes **Total Marks: 60**

Q1 Develop a vehicle management system using abstract classes and exception handling. The system should handle different vehicle types and their operations. Create an abstract class called Vehicle with attributes like registrationNumber, brand, and year. Implement a constructor and an abstract method calculateMileage(). Create a concrete class Car that extends Vehicle with an additional attribute fuelCapacity. Implement calculateMileage() using the formula totalDistance / fuelConsumed. Handle exceptions for zero or negative fuel consumption using a custom exception InvalidFuelConsumptionException. **(12M)**

```
Mileage: 20.0 km/l
```

Q2 Create a generic class called StudentPair to store a pair of Student objects. Implement methods to get and set student values. Create two instances of the StudentPair class using different Student objects. Set values for the students in each pair with names and ages of your choice. Display the names and ages of both students in each pair using the getName() and getAge() methods. Use the setFirstStudent() and setSecondStudent() methods to update the values of the students in each pair. Display the updated names and ages of the students in each pair. Ensure that your code handles null objects appropriately (you can assume that the Student objects passed to the StudentPair constructor and the setFirstStudent() and setSecondStudent() methods will not be null). **(12M)**

```
Students in pair1:
First Student: Alice, Age: 20
Second Student: Bob, Age: 22

Students in pair2:
First Student: Carol, Age: 19
Second Student: Dave, Age: 21

Updated students in pair1:
First Student: Eve, Age: 23
Second Student: Frank, Age: 24
```

Q3 Implement a Java program that calculates the sum of numbers from 1 to 100 using multi-threading. The program consists of two threads: a main thread and a child thread. The child thread performs the summation and notifies the main thread when it is done. The main thread waits for the notification from the child thread and then displays the calculated sum. Provide a code snippet that demonstrates the synchronization and inter-thread communication using wait() and notify(). Additionally, explain how the main thread and child thread interact with each other to ensure synchronization and proper calculation of the sum." **(12M)**

user enters the total number of threads to be executed and number of times of execution of each thread also should be entered by the user. To make one thread wait for other threads to terminate you can use join method effectively.

[12]

Q4. Assume a scenario of movie ticket booking system where two customers are trying to book a ticket for a same movie. Both the customers are trying to book the tickets in the same time. Think that only one seat is available in the theater and both the customers are asking for the same ticket. Design a Java program using thread synchronization so that only one ticket is allocated to one customer at a time. To design program, take a class named as "Movie_Ticket" that implements runnable. Take two variables such as total_seats_available and seats_requested. Create two threads t1 and t2. When t1 enters the run method, it checks the "total_seats_available" and allot it to the first customer. After the allocation of the seat the thread t1 enters try{} block inside the run method and goes to sleep for 5 seconds. During this sleep duration the ticket will be printed in the printer. The program can only compile successfully compiled by using interthread communication.

Sample Output:

Available seats= **1**

1 Seat reserved **for** First Person

Available Seats= **0**

Sorry! No seats are available

[12]

Q5. In your weather application, imagine the current temperature is displayed in Celsius, and you want to switch it to Fahrenheit. Design the button click required to achieve this conversion. To switch the temperature unit from Celsius to Fahrenheit in the weather application, you would need to click on the "Convert to Fahrenheit" button.

[12]

QP MAPPING

Q. No.	Module Number	CO Mapped	PO Mapped	PEO Mapped	PSO Mapped	Marks
Q1	2	3,4	1,2,3,4,5,6,7	PEO3,4	PSO2,3	12
Q2	4	3,4,5	1,2,3,4,5,6,8	PEO3,4	PSO2,3	12
Q3	5	3,4,5	1,2,3,4,5,6,8	PEO3,4	PSO2,3	12

UNIVERSITY
Apply Knowledge, Improve Life!

QUESTION PAPER

Name of the Examination: WIN 2022-2023 - FAT

Course Code: CSE2005 **Course Title:** Object Oriented Programming

Set number: 3 **Date of Exam:** 20/06/2023 (FN) (D1)

Duration: 120 Minutes **Total Marks: 60**

Instructions:

1. Assume data wherever necessary.
2. Any assumptions made should be clearly stated.

Q1. Write a Java code that demonstrates the usage of the Book and Loan classes. Create an instance of the Book class with a title "Java Programming" by "John Smith" published in the year 2022. Then, create an inner class object of borrow for this book, set the borrower name as "Alice", borrow date as "2023-05-01", and due date as "2023-05-15". Finally, display the book details along with the borrower's name, borrowed date, and due date. Consider the class names and methods as enlisted below.

1. Class Names: Book: Represents a book and contains the information about the title, author and publication. Inner Class: Burrow: Represents a book burrow transaction and contains methods to record borrower information, loan date, and due date.
2. Methods: For Book Class : getTitle(): Returns the title of the book. getAuthor(): Returns the author of the book. getPublicationyear(): Returns the publication year of the book. For Borrow Class (Inner): SetName: Sets the name of the burrower. setDate(): returns burrowed date, dueDate(): Returns the due date for return. [12]

Q2. Write a Java code that demonstrates the usage of the Box class with different data types. Create two instances of the Box class, one for storing integers and another for storing strings. Add some integers and strings to their respective boxes. Finally, call the printContents method for each box to display their contents. Consider the class names and method names as enlisted below.

Class name: Box<T>

Methods: addItem(): adds and item of type T to the box and PrintContent(): prints the content of the box [12]

Q3.

```
static class IncrementCounter {
int counter;
void increment () {
 counter = counter+1;
                    }
int readCount() {
return count;
                    }
          }
```

In this question a few lines of codes are mentioned. Read them carefully. Write down a thread class that will repeatedly call the increment () in an object of type IncrementCounter. Ensure that the object should be a shared global variable. Create as many as threads and wait for all the threads to execute and terminate. Then, print the final value of the counter. Design the program in such a manner that the