

Indian Institute of Technology, Madras - BS in Data Science and Applications

Notations :

- 1.Options shown in green color and with ✓ icon are correct.
- 2.Options shown in red color and with ✗ icon are incorrect.

Question Paper Name :

IIT M FOUNDATION AN4 EXAM QPF4 16
JULY 2023

Subject Name :

2023 July: IIT M FOUNDATION AN4 EXAM
QPF4

Creation Date :

2023-07-10 18:54:05

Duration :

240

Total Marks :

705

Display Marks:

Yes

Share Answer Key With Delivery Engine :

Yes

Actual Answer Key :

Yes

Calculator :

Scientific

Magnifying Glass Required? :

No

Ruler Required? :

No

Eraser Required? :

No

Scratch Pad Required? :

No

Rough Sketch/Notepad Required? :

No

Protractor Required? :

No

Show Watermark on Console? :

Yes

Highlighter :

No

Auto Save on Console?

Yes

Change Font Color :

No

6406531929152. ✓ No

Question Number : 248 Question Id : 640653577689 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Short Answer Question

Using the least square approximation, find the length of the projection vector p of b onto the column space of A . Enter the answer correct to two decimal places.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

3.12 to 3.16

Java

Section Id :	64065339063
Section Number :	13
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	16
Number of Questions to be attempted :	16
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0

Sub-Section Number : 1

Sub-Section Id : 64065382533

Question Shuffling Allowed : No

Is Section Default? : null

Question Number : 249 Question Id : 640653577695 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : PROGRAMMING CONCEPTS USING JAVA (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE [TOP](#) FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531929171.  YES

6406531929172.  NO

Sub-Section Number : 2

Sub-Section Id : 64065382534

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 250 Question Id : 640653577696 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which among the following statements is/are **true** about activation records?

Options :

6406531929173. ✖ All the variables on the stack (in any activation record) are in their scope.

6406531929174. ✖ Only the variables present in the topmost activation record of the stack are in their lifetime

6406531929175. ✔ Every activation record has a control link that points to start of previous record.

6406531929176. ✔ Every activation record has a return value link that points to where to store return value.

Question Number : 251 Question Id : 640653577698 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the code given below that checks whether two books are the same. Method `equals` is overridden to compare two `Book` objects as follows. If two books have the same author name and book name, then they are the same. Based on the given information, answer the question that follows.

```
class Book{
    private String bName; // Book name
    private String aName; // Author name

    //Constructor to initialize instance variables

    public String toString() {
        return bName;
    }
    public boolean equals(Object obj) {
        // CODE BLOCK
    }
}

public class Test {
    public static void main(String[] args) {
        Book b1 = new Book("ABC", "XYZ");
        Book b2 = new Book("ABC", "XYZ");
        if(b1.equals(b2))
            System.out.println(b1+" "+ b2 +" are same");
        else
            System.out.println(b1+" "+ b2 +" are different");

    }
}
```

Choose the correct option(s) to fill in place of `CODE BLOCK` so that the output is:

ABC, ABC are same

Options :

```
if(obj instanceof Book) {
    if(this.bName.equals(obj.bName) && this.aName.equals(obj.aName))
        return true;
}
return false;
```

6406531929181. ✖

```
if(this.bName.equals(obj.bName) && this.aName.equals(obj.aName))
    return true;
return false
```

6406531929182. ✖

```
if(obj instanceof Book) {  
    Book bk = (Book) obj;  
    if(this.bName.equals(bk.bName) && this.aName.equals(bk.aName))  
        return true;  
}  
return false;
```

6406531929183. ✓

```
if(obj instanceof Book) {  
    Book bk = obj;  
    if(this.bName.equals(bk.bName) && this.aName.equals(bk.aName))  
        return true;  
}  
return false;
```

6406531929184. ✖

Question Number : 252 Question Id : 640653577701 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the code given below.

```
class Rectangle{
    private int sides;
    public Rectangle(int s){
        sides = s;
    }
    public final void area(){
        System.out.println("area of rectangle");
    }
}

class Pentagon extends Rectangle {
    public Pentagon(int n){
        super(n);
    }
    public final void area(){ // LINE 1
        System.out.println("area of pentagon");
    }
}

public class Test{
    public static void main(String[] args){
        Rectangle r = new Pentagon(4); // LINE 2
        r.sides = 5; // LINE 3
        r.area();
    }
}
```

Choose the correct option(s).

Options :

6406531929193. ✓ LINE 1 generates compilation error because the method `area()` cannot be overridden.

6406531929194. ✗ LINE 2 generates compilation error because a variable of type `Rectangle` cannot refer to an object of type `Pentagon`.

6406531929195. ✓ LINE 3 generates compilation error because instance variable `sides` cannot be accessed in class `Test`.

6406531929196. ✗

This code generates the output:
area of pentagon

Question Number : 253 Question Id : 640653577707 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the code given below.

```
interface Moveable{
    void move();
}
class Fan implements Moveable{
    public void move() {
        System.out.println("Move and circulate air inside the room");
    }
}
class Cooler{
    public Fan getBlower() {
        return new Blower();
    }
    private class Blower extends Fan implements Moveable{
        public void move() {
            System.out.println("Move and throws air at high pressure");
        }
    }
}

public class PrivateTest {
    public static void main(String[] args) {
        //CODE BLOCK
    }
}
```

Choose the correct option(s) to be filled in place of CODE BLOCK so that the output is:

Move and throws air at high pressure

Options :

6406531929217. ✓


```
Cooler obj1 = new Cooler();  
Moveable obj2 = obj1.getBlower();  
obj2.move();
```

```
Cooler obj1 = new Cooler();  
Blower obj2 = obj1.getBlower();  
obj2.move();
```

6406531929218. ✖

```
Moveable obj1 = new Blower();  
obj1.move();
```

6406531929219. ✖

```
Cooler obj1 = new Cooler();  
Fan obj2 = obj1.getBlower();  
obj2.move();
```

6406531929220. ✔

Sub-Section Number : 3

Sub-Section Id : 64065382535

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 254 Question Id : 640653577697 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Match the following terms with their descriptions/properties.

A. Activation Record	1. Compatibility of interfaces
B. Dynamic lookup	2. Restricting modification of data to the methods of the object only
C. Subtyping	3. Reuse of implementations
D. Inheritance	4. Stores the local variables
E. Encapsulation	5. Choice of method implementation determined at run-time
	6. Stores the instance variables

Options :

6406531929177. ✖ A-6, B-5, C-3, D-1, E-2

6406531929178. ✔ A-4, B-5, C-1, D-3, E-2

6406531929179. ✖ A-4, B-2, C-3, D-1, E-5

6406531929180. ✖ A-6, B-2, C-1, D-3, E-5

Question Number : 255 Question Id : 640653577699 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the code given below.

```
class Vehicle{
    public void move() {
        System.out.println("Moves");
    }
}
class Bike extends Vehicle{
    public void seating() {
        System.out.println("Bike seating");
    }
}
class Car extends Bike{
    public void seating() {
        System.out.println("Car Seating");
    }
    public void safety() {
        System.out.println("Offers safety");
    }
}
public class Test{
    public static void main(String[] args) {
        Vehicle v = new Bike();
        Bike b = new Car(); // LINE 1
        v.move();
        ((Bike)v).seating(); // LINE 2
        b.safety(); // LINE 3
    }
}
```

Choose the correct option.

Options :

6406531929185. ✖ LINE 1 generates compilation error because a variable of type Bike cannot refer to an object of type Car.

6406531929186. ✖ LINE 2 generates compilation error because a variable of type Vehicle cannot be type casted to an object of type Bike.

6406531929187. ✔ LINE 3 generates compilation error because the method `safety()` is not defined in class Bike.

This code generates the output:

Moves
Bike seating
Offers safety

6406531929188. ✖

Question Number : 256 Question Id : 640653577700 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the code given below.

```
class Player{
    public void play(){
        System.out.println("Player plays");
    }
    public void play(String n){
        System.out.println("Player is " + n);
    }
}
class Captain extends Player{
    public void play(String s){
        System.out.println("Captain is " + s);
    }
}
public class Test{
    public static void main(String[] args){
        Player p = new Captain(); // LINE 1
        p.play();
        p.play("siva"); // LINE 2
    }
}
```

Choose the correct option.

Options :

LINE 1 generates compilation error because a variable of type Player cannot refer to an object of type Captain.

6406531929189. ✖

This code generates the below output followed by runtime Error at LINE 2 because there is ambiguity in which play() method is being invoked.

Player plays

6406531929190. ✖

This code generates the output:

Player plays

Player is siva

6406531929191. ✖

This code generates the output:

Player plays

Captain is siva

6406531929192. ✔

Question Number : 257 Question Id : 640653577702 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class Employee{
    private String name;
    private double salary;
    public Employee(String n, double s) {
        name = n;
        salary = s;
    }
    public String toString() {
        return "name = " + name + ", salary = " + salary;
    }
}
class Manager extends Employee{
    String dateOfPromotion;
    public Manager(String name, double salary, String dop) {
        //LINE-1
        dateOfPromotion = dop;
    }
    public String toString() {
        return super.toString()+"", dateOfPromotion = "+dateOfPromotion;
    }
}
public class Test {
    public static void main(String[] args) {
        Employee obj = new Manager("Suyan", 500000.00, "01/06/2021");
        System.out.println(obj);
    }
}
```

Choose the correct option to be filled in place of LINE-1 so that the output is:

name = Suyan, salary = 500000.0, dateOfPromotion = 01/06/2021

Options :

6406531929197. ✖ this(name, salary);

6406531929198. ✔ super(name, salary);

6406531929199. ✖ Employee(name, salary);

6406531929200. ✖ super.Employee(name, salary);

Question Number : 258 Question Id : 640653577703 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
public class SwitchTest {  
    public static void show() {  
        for(int i=0;i<3;i++) {  
            switch(i) {  
                case 1:  
                    System.out.print("1 ");  
                case 2:  
                    System.out.print("2 ");  
                case 3:  
                    System.out.print("3 ");  
                default:  
                    System.out.print("Default case");  
                    break;  
            }  
            System.out.println();  
        }  
    }  
    public static void main(String[] args) {  
        show();  
    }  
}
```

What will the output be?

Options :

1 2 3
2 3

6406531929201. ✖

Default case
1 2 3 Default case
2 3 Default case

6406531929202. ✔

Default case

1

2

6406531929203. ✖

Default case

1 2 3

2 3

6406531929204. ✖

Question Number : 259 Question Id : 640653577704 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the code given below.

```
abstract class Phone{
    abstract void basic_features();
    abstract void additional_features();
}
abstract class KeypadPhone extends Phone{ // LINE 1
    public void basic_features() {
        System.out.println("Calling/Messaging");
    }
}
class SmartPhone extends KeypadPhone{ // LINE 2
    public void additional_features() {
        System.out.println("Touch screen/Camera");
    }
}
public class Test {
    public static void main(String[] args) {
        KeypadPhone obj1 = new SmartPhone();
        Phone obj2 = new SmartPhone();
        obj1.basic_features();
        obj2.additional_features(); // LINE 3
    }
}
```

Choose the correct option.

Options :

This code generates the output:

Calling/Messaging

6406531929205. ✓ Touch screen/Camera

LINE 1 generates compilation error because method `additional_features()` is not overridden in `KeypadPhone` class.

6406531929206. ✖

LINE 2 generates compilation error because method `basic_features()` is not overridden in `SmartPhone` class.

6406531929207. ✖

LINE 3 generates compilation error because object `obj2` should be cast to its child class `SmartPhone`.

6406531929208. ✖

Question Number : 260 Question Id : 640653577705 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the code given below.

```
interface Appraisable{
    public default void isAppraisable() {
        System.out.println("Yes this employee appraisable");
    }
}
class Employee{
    int leaves;
    public Employee(int num) {
        leaves = num;
    }
    public void isAppraisable() {
        if(leaves >= 20)
            System.out.println("This employee not appraisable");
    }
}
class Manager extends Employee implements Appraisable{
    public Manager(int num) {
        super(num);
    }
}
public class InterfaceTest {
    public static void main(String[] args) {
        Appraisable obj1 = new Manager(10);
        obj1.isAppraisable();
        Appraisable obj2 = new Manager(20);
        obj2.isAppraisable();
    }
}
```

Choose the correct option.

Options :

6406531929209. ✖ This program generates no output.

6406531929210. ✖ This code generates the output:
This employee not appraisable
Yes this employee appraisable

6406531929211. ✔ This code generates the output:
This employee not appraisable

This code generates the output:
Yes this employee appraisable
This employee not appraisable

6406531929212. ✖

Question Number : 261 Question Id : 640653577706 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the code given below.

```
class Doctor {
    private String name = "****";
    private static String hospital_group = "ABC";
    public Doctor(){
        name = "**";
    }
    public Doctor(String n){
        name = n;
    }
    public String toString() {
        return "name = " + name + ", hospital group = " + hospital_group;
    }
}

public class VarTest{
    public static void main(String[] args) {
        Doctor obj1 = new Doctor();
        Doctor obj2 = new Doctor("XYZ");
        System.out.println(obj1);
        System.out.println(obj2);
    }
}
```

What will the output be?

Options :

name = null, hospital group = ABC
name = XYZ, hospital group = ABC

6406531929213. ✖

6406531929214. ✖
name = **, hospital group = null
name = XYZ, hospital group = ABC

6406531929215. ✖
name = ****, hospital group = ABC
name = XYZ, hospital group = ABC

6406531929216. ✔
name = **, hospital group = ABC
name = XYZ, hospital group = ABC

Question Number : 262 Question Id : 640653577709 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the code given below.

```
class Person{
    private String name, location;
    private int age;
    public void setLocation(String location) {
        this.location = location;
    }
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public Person(String name, int age, String location) {
        this.name = name;
        this.age = age;
        this.location = location;
    }
    public Person(Person p) {
        this(p.name, p.age, "Hyderabad");
    }
    public String toString() {
        return name + ", " + age + ", " + location;
    }
}

public class CopyConTest {
    public static void main(String[] args) {
        Person p1 = new Person("ABC", 23);
        Person p2 = new Person(p1);
        p1.setLocation("Mumbai");
        System.out.println(p1);
        System.out.println(p2);
    }
}
```

What will the output be?

Options :

null, 0, Mumbai

6406531929225. ✖ ABC, 23, Hyderabad

ABC, 23, Mumbai

6406531929226. ✖ ABC, 23, Mumbai

6406531929227. ✖

ABC, 23, Hyderabad
ABC, 23, Hyderabad

ABC, 23, Mumbai

6406531929228. ✓ ABC, 23, Hyderabad

Question Number : 263 Question Id : 640653577710 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class Animal{
    String name;
    int legs;
    public Animal(String n, int l){
        name = n;
        legs = l;
    }
    ***-----***
    * CODE SEGMENT *
    ***-----***
}
public class Test{
    public static void main(String[] args){
        Animal a1 = new Animal("Dogs", 4);
        Animal a2 = new Animal("Birds", 2);
        System.out.println(a1 + "\n" + a2);
    }
}
```

Choose the correct option to fill in the CODE SEGMENT so that the output is:

Dogs : 4
Birds : 2

Options :

6406531929229. ✓


```
public String toString(){  
    return name + " : " + legs;  
}
```

```
        public String toString(Object ob){  
            return ob.name + " : " + ob.legs;  
        }  
6406531929230. ✖
```

6406531929231. ✖ No additional code is required in place of CODE SEGMENT.

6406531929232. ✖ This output will not be printed because Java throws an error when an object is tried to be printed using `System.out.println`.

Sub-Section Number : 4

Sub-Section Id : 64065382536

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 264 Question Id : 640653577708 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
interface Bookable{
    public void booked();
    public void rejected();
}
class TicketBookingSite{
    public static int maxTickets = 100;
    Bookable obj;
    // Constructor to initialize Bookable obj
    public void startBooking(int tickets) {
        if(tickets <= maxTickets) {
            obj.booked();
            maxTickets -= tickets;
        }
        else
            obj.rejected();
    }
}

class Audience implements Bookable{
    String name;
    int noTickets;
    // Constructor initialize name and noTickets
    public void getTickets() {
        TicketBookingSite site = new TicketBookingSite(this);
        site.startBooking(noTickets);
    }
    public void booked() {
        System.out.println("You got "+this.noTickets+" tickets");
    }
    public void rejected() {
        System.out.println("Tickets sold out");
    }
}

public class CallBack {
    public static void main(String[] args) {
        Audience a1 = new Audience("A", 34);
        Audience a2 = new Audience("B", 67);
        Audience a3 = new Audience("C", 34);
        a1.getTickets();
        a2.getTickets();
        a3.getTickets();
    }
}
```

Choose the correct option.

Options :

6406531929221. ✖ This program generates no output.

This program generates the output:
You got 34 tickets
Tickets sold out
You got 34 tickets
6406531929222. ✓

This program generates the output:
You got 34 tickets
You got 34 tickets
6406531929223. ✖

This program generates the output:
You got 34 tickets
Tickets sold out
6406531929224. ✖

AppDev2

Section Id :	64065339064
Section Number :	14
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065382537
Question Shuffling Allowed :	No