

## CS-3391 OBJECT ORIENTED PROGRAMMING

### Question Bank

#### UNIT – 1

##### PART – A

- 1 Express what is meant by Object Oriented Programming.
- 2 Compare class and object.
- 3 List the core OOP's concepts.
- 4 Tabulate the difference between C++ and Java.
- 5 Discuss what is meant by abstraction.
- 6 Describe about Encapsulation, Inheritance and Polymorphism.
- 7 Point out the justification of the statement "Java is platform independent".
- 8 Express a Java programming structure to display "Hello World".
- 9 List the various access specifiers supported by OOPS.
- 10 Illustrate constructors in Java.
- 11 Express a simple Java Program to find the given number is Prime or not.
- 12 Write down the characteristics of objects.
- 13 Illustrate with example what is meant by parameter passing constructor.
- 14 Quote the purpose of finalize methods.
- 15 List out the type of Arrays.
- 16 Illustrate the working of Java Virtual Machine (JVM).
- 17 Define static variable and static method.
- 18 Explain what is meant by Java package.
- 19 Illustrate how to import a single package.
- 20 Summarize any four Java doc comments. List any four common browsers.

##### PART – B

1. I. Explain OOPS and its features. (7)  
ii. Summarize about the usage of constructor with an example using Java. (6)
- 2 i. What is class? How do you define a class in Java? (7)  
ii. Examine the use of inheritance and class hierarchy. (6)
- 3 i. Define polymorphism. (7)  
ii. Describe variables and operators in Java. (6)
4. Define and explain the control flow statements in Java with suitable examples. (13)
5. What is meant by constructor? Discuss the types of constructor with example. (13)
- 6 i. Analyse and Develop a simple Java program to sort the given numbers in increasing order (7)  
ii. Write a Java program to reverse the given number. (6)

7. i. Classify the characteristics of Java. (6)
- ii. Illustrate the working principles of Java Virtual Machine. (4)
- iii. Show with an example the structure of Java Program (3)
8. i. Summarize about access specifier in Java. (7)
- ii. Describe the term static fields and methods and explain its types with example. (6)
9. i) Define Arrays. What is array sorting and explain with an example (7)
- ii) Tabulate and explain documentation comments in Java. (6)
10. Illustrate what is meant by package? How its types are created and implemented in Java. (13)
11. Write the techniques to design classes in Java using JavaDoc. (13)
12. Explain with example passing objects as parameters to methods and returning objects from methods in Java. (13)
13. Explain packages in Java with example. (13)
14. Interpret with an example what is method overloading and method overriding. (13)

### **PART – C**

1. Illustrate a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, and type of EB connection (i.e. domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- i. First 100 units - Rs. 1 per unit
- ii. 101-200 units - Rs. 2.50 per unit
- iii. 201 -500 units - Rs. 4 per unit
- iv. > 501 units - Rs. 6 per unit (15) –

2. Explain a Java program to find a smallest number in the given array by creating one dimensional array and two dimensional array using new operator. (15)

3. Explain class hierarchy and explain its types with suitable examples. (15)

4. Express a Java application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10% of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary. (15)

### **UNIT – II**

#### **PART – A**

1. Examine the importance of inheritance.
2. Summarize the characteristics of constructor function.
3. What is a default constructor? Illustrate.
4. Identify what are the two ways of using super keyword.

5. What modifiers may be used with top-level class?
6. Give how protected members in a subclass can be accessed in Java.
7. Show what methods are provided by the object class.
8. Point out the conditions to be satisfied while declaring abstract classes.
9. Give the use of final keyword.
10. Illustrate what is protected visibility.
11. Define interface and write the syntax of the interface.
12. Express what is Dynamic Binding.
13. Assess what is a cloneable interface and how many methods does it contain.
14. Describe whether you can have an inner class inside a method and what variables can you access.
15. Differentiate between abstract class and interface.
16. Define: Package.
17. Give the role of clone () method in Java.
18. Point out what are inner class and anonymous class.
19. What is the use of Array List class?
20. Summarize any two string handling methods in Java.

## **PART – B**

1. i. Describe in detail about inheritance. (7)  
 ii. Write a program for inheriting a class. (6)
2. i. Illustrate what is super and subclass in Java. (7)  
 ii. With an example, illustrate how the objects from sub class are inherited by the super class. (6)
3. Examine how to control top level and member level access for the members of the class. (13)
4. Illustrate with an example how passing objects as parameters to methods and returning objects from methods in Java. (13)
5. Describe in brief about object class and its methods in Java with suitable example. (13)
6. i. Discuss the concept of abstract class. (7)  
 ii. Give a program for abstract class with example. (6)
7. i. Explain briefly on final keyword. (7)  
 ii. Explain the concept of abstract class with an example. (6)
8. i. Describe what is meant by interface. (7)  
 ii. How is interface declared and implemented in Java. Give example. (6)
9. i. Differentiate classes with interface with suitable examples. (7)  
 ii. Express a Java program for extending interfaces. (6)
10. Define inner classes. How to access object state using inner classes? Give an example. (13)
11. i. Explain with an example what is meant by object cloning? (7)  
 ii. Summarize in detail about inner class with its usefulness. (6)
12. Analyse and write a Java program using ArrayList classes and object for the following operations.
  - i. Push (7)
  - ii. Pop (6)

13. Analyse with an example, how string objects are created. How it can be modified?(13)

14. Illustrate String handling class in Java with example. (13)

15. Explain in detail about Package with an Example Program. (13)

## **PART – C**

1. Illustrate a program to perform string operations using ArrayList. Write functions for the following

Append - add at end

Insert – add at particular index Search

List all string starts with given letter "a". (15)

2. Assess and write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Specify the instance variable and methods for each class. The private instance variables of Quadrilateral should be the x-y coordinate pairs for the four end points of the quadrilateral.

Write a program that instances objects of your classes and outputs each objects area (except Quadrilateral) (15)

3. Consider a class student .Inherit this class in UG Student and PG Student. Also inherit students into local and non-localstudents. Define five Local UG Students with a constructor assuming all classes have a constructor. (15)

4. Express a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape. (15)

## **UNIT – III**

### **PART – A**

1. Interpret what is an Exception. What is its use?

2. Predict what function does terminate and unexpected handlers call.

3. What is re-throwing an expression?

4. Define uncaught exception.

5. Summarize the task performed by exception handling.

6. Differentiate exception and error.

7. Classify the exception types with example.

8. Draw the exception hierarchy.

9. What are the two methods available in stack trace elements?

10. Express the advantages of using exception handling.

11. What are three types of I/O streams?

12. Show what is the purpose of the finally clause of a try-catch-finally statement?

13. Explain how to create custom exception.

14. List the any five byte stream class.

15. Illustrate any four character stream class.

16. Point out the syntax of buffered reader to connect to the keyboard.

17. What are streams? What are their advantages?
18. Write a Java program to demonstrate the use of readlinemethod.
19. Illustrate a Java application using a printwriter class to handle console output.
20. Define: Thread.

## **PART – B**

1. Discuss in detail about exception handling constructs and write a program to illustrate Divide by zero exception. (13)
2. Describe the following concepts with example.
  - i. Try-catch-throw paradigm. (7)
  - ii. Exception specification. (6)
3. Describe about the syntax of catch and re-throw an exception with an example. (13)
4. (i) Point out the importance of exception hierarchy. (7)  
 (ii) Explain on stack trace elements give example (6)
5. Tabulate any five classes to support exception handling in Java with an example for each. (13)
6. i. Summarize what is finally class. How to catch exception? Write an example. (7)  
 ii. Give short notes on Java built in exceptions (6)
7. Explain the following in detail with example program
  - i. Checked Exception (7)
  - ii. Unchecked exception (6)
8. i. Classify the errors and exception in Java. (7)  
 ii. Illustrate about multiple catching exceptions with example. (6)
9. Summarize the following with example program
  - i. Arithmetic exception (5)
  - ii. Arrayoutofbound exception (4)
  - iii. Stringindexoutofbound exception (4)
10. i. Express a program to read and count the characters from files. (7)  
 ii. Illustrate a Java program to transfer the content of one file to another file. (6)
11. Discuss briefly about the features
  - i. Byte streams input/output (7)
  - ii. Character streams input/output (6)
12. Explain the following with example
  - i. Reading console input (7)
  - ii. Writing console output. (6)
13. i. Identify a Java program to read characters from the console. (7)  
 ii. Identify a Java program to read strings from the console. (6)
14. Illustrate in brief about
  - i. Reading from a file. (7)
  - ii. Writing in a file. (6)
15. Explain in detail about Thread and its types. (13)

## **PART – C**

1. Implement a Java program for user defined exception handling. (15)
2. i. Custom exception has been created in the code given below. Correct and evaluate the code (15)

```

Class myexception extends Exception
{
Myexception(string s)
{
super(s)
}
}
Class excep
{
Public static void main (String args [])
{` if (args [0] == "Hello") System.out.println("String is right");
else try
{
Throw new myexception("Invalid string");
}
Catch(myexception ex)
{
System.out.println(ex.gemessage());
}
}
}

```

ii. The program calculates sum of two numbers inputted as command-line arguments. When will it give an exception? Class excep

```

{
Public static void main( String []args)
{
try{
int n= Integer.parseInt(arg[0]); int n1=Integer.parseInt(arg[1]); int n2=n+n1;
System.out.println("Sum is:" +n2);
}
Catch(NumberFormatException ex)
{
System.out.println(ex);
}
}
}

```

3. Illustrate the Java program to concatenate the two files and produce the output in the third file. (15)

4. Deduce a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes. (15)

## **UNIT – IV**

### **PART - A**

1. Give the properties of thread.
2. Show the different states in thread.

3. Why synchronization is required in thread?
4. Select any four thread constructor.
5. What is the need for thread?
6. List the importance of thread constructor.
7. Give the idea to achieve thread synchronization in Java.
8. Define multithreading.
9. Give the life cycle of thread.
10. Show how virtual machine is used in generic programming.
11. Write down the need for generic programming.
12. Define generic class.
13. Illustrate a simple generic class with an example.
14. Summarize the advantages of generic programming.
15. What are the parts of synchronizers that are often needed?
16. Give the methods used for inter thread communication.
17. Classify what are three ways in which a thread can enter the waiting state?
18. Illustrate what is daemon thread and which method is used to create the daemon thread.
19. Differentiate between yielding and sleeping.
20. Illustrate what is thread group.

## **PART – B**

1. Describe in detail about multithread programming with (13) example.
2. i. Differentiate multithreading and multitasking. (7)
- ii. Describe the properties of thread in detail. (6)
3. Summarize the two types of thread implementation supported by Java. Give example. (13)
4. i. Illustrate the concept of synchronization in thread. (7)
- ii. Write a Java code for reader writers problem. (6)
5. i. Describe how to implement runnable interface for creating and starting threads. (7) –
- ii. Define threads. Describe in detail about thread life cycle. (6)
6. i. Explain what is inter-thread communication? List out the methods used for it. (7)
- ii. Explain inter-thread communication using producer-consumer problem. (6)
7. Summarize the following
  - i. Thread priorities (7)
  - ii. Daemon thread (6)
8. Explain the following
  - i) States of a thread with a neat diagram (7)
  - ii) Explain how threads are created in Java (6)
9. i. Illustrate the motivations of generic programming. (7)
- ii. Develop a program to show generic class and methods with example. (6)
10. i. Describe in detail about bounded types with suitable example. (7)
- ii. List the inheritance rules for generic types with example. (6)
11. i. Give the limitations of generic programming. (7)
- ii. Explain any two restrictions of generic programming in detail with suitable example. (6)

- 12 Describe the following
  - i. Generic class (7)
  - ii. Generic method (6)
13. Illustrate generic code and the virtual machine with suitable example. (13)
14. Summarize thread group. How to implement the thread group. Explain it with example. (13)

### **PART – C**

1. Express multithreading for an sample sequence of strings with a delay of 1000 millisecond for displaying it using Java threads. (15) –
2. Deduce a Java program to perform the following tasks using three different threads. Each thread will be responsible for its own task only. Among these three threads one will find the average number of the input numbers, one will be responsible for finding the Maximum number from the input array of numbers, and one will be responsible for finding the Minimum number from the input array of numbers. (15)
3. Express a simple generic class example with two type parameters. so that we can define two types of parameters called U & V, separated by ",". (15)
- 4 Assess an example program in Java on how to implement bounded types (extend superclass) with generics. (15)

### **UNIT – V**

#### **PART – A**

1. List out some system colors available in Java and their purpose.
2. Give the role of layout manager. Which layout is default in Java? Define Border Layout.
3. Give the steps needed to show a Frame.
4. Analyse the function of
  - I. Set Layout
  - ii. Flow Layout
5. List the features of Swing.
6. Name any four event of a button component.
7. List the various mouse events supported by Java.
8. Recommend what method can be used for changing case of characters.
9. Illustrate the swing components.
10. Differentiate between Swing and AWT.
11. Express what is an event and what are the models available for event handling.
12. List the difference between scrollbar and scroll pane.
13. Differentiate between a Choice and a List.
14. Quote how can you create your own GUI components?
15. Analyse what is the purpose of the enable Events () method?
16. Write a program to print the names of all fonts on your system.
17. Show the methods of frame class.
18. Show the structure of AWT Event Hierarchy.
19. Give what is JRadioButton and its constructor.



20. Show what method can be used for changing font of characters?

## **PART – B**

- 1.i. Describe in detail about working with 2D shapes in Java. (7)
- ii. Identify a Java program to illustrate Mouse Events. (6)
- 2.i..Describe in detail about swing Components. (7) ii.Describe the types of layout management. (6)
3. Summarize in detail about graphics programming. (13)
- 4.I.Explain how an application can respond to events in Java? Write the steps and the example. (7)
- ii. Explain the adapter class using example. (6)
5. What is meant by event handling? Analyze and write a simple calculator using mouse events that restrict only addition, subtraction, multiplication and division. (13)
6. Tabulate the controller design pattern and components of swing briefly. (13)
- 7.i.Illustrate what is layout management? State the various types of layout supported by Java? Which layout is default one? (7)
- ii.Examine the basic of event handling. (6)
8. Explain with an example program and discuss in detail about Mouse listener and Mouse Motion Listener. (13)
- 9.i.Illustrate the methods available in graphics for COLOR. (7)
- ii.Examine the methods available to draw shapes. (6)
- 10 i. Explain on AWT Event Hierarchy (7)
- ii. Explain about Semantic and Low-Level Events (6)
11. List the characteristics of Model View Design (MVC) patterns. Explain the advantage of MVC and methods MVC. (13)
- 12.i.List the types of adjustment events in scrollbar. (7)
- ii.Explain and write a program to demonstrate the usage of Scroll bar. (6)
- 13 Examine the following in detail
- i. Handling a TextField. (7)
- ii. Using a TextArea. (6)

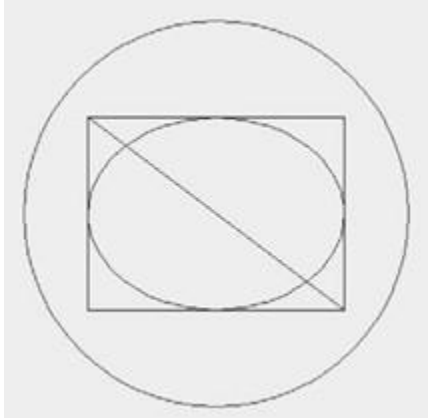
## **PART – C**

1. Illustrate a Java program to implement the following

Create four check boxes. The initial state of the first box should be in checked state. The status of each check box should be displayed. When we change the state of a check box, the status should be displayed and updated. (15)

2. Express a Java program to display the following picture as (15)

Output.



3. Explain a Java program for event handling using ActionListener interface (15)



4. Recommend a Java swing with one button and adding it on the JFrame object inside the main() method. (15)