|  |
| --- |
| What is multithreading?  Multithreading in Java is **a process of executing multiple threads simultaneously**. A thread is a lightweight sub-process, the smallest unit of processing. |
| Sketch the life cycle of a thread.  life-cycle-of-thread-in-java-.jpg |
| What are the states of a thread?   1. New 2. Runnable 3. Running 4. Blocked (Non-runnable state) 5. Dead |
| Write note on start().  The **java.lang.Thread.start()**method causes this thread to begin execution, the Java Virtual Machine calls the run method of this thread.The result is that two threads are running concurrently: the current thread (which returns from the call to the start method) and the other thread (which executes its run method). |
| Write note on run().  The run() method of thread class is called if the thread was constructed using a separate Runnable object otherwise this method does nothing and returns. When the run() method calls, the code specified in the run() method is executed. |
| Write note on wait().  In Java, the wait() method is used to pause the execution of a thread until another thread signals that it can resume. When a thread calls wait() on an object, it releases the lock on the object and waits until another thread calls notify() or notifyAll() on the same object. |
| Write note on notify().  notify() wakes up a single thread that is waiting on this object's monitor. If many threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation. |
| Write note on notifyAll().  notifyAll() wakes up all threads that are waiting on this object's monitor. A thread waits on an object's monitor by calling one of the wait methods. The awakened threads will not be able to proceed until the current thread relinquishes the lock on this object. |
| What is a stream?   A stream can be defined as a sequence of data. Java performs I/O through Streams. The InputStream is used to read data from a source and the OutputStream is used for writing data to a destination. |
| Differentiate byte and character stream.  Byte streams process data byte by byte (8 bits). For example, FileInputStream is used to read from the source and FileOutputStream to write to the destination.  In Java, characters are stored using Unicode conventions. Character stream automatically allows us to read/write data character by character. For example, FileReader and FileWriter are character streams used to read from the source and write to the destination. |
| List Byte and character stream classes.  ByteStreamVsCharacterStreamInJava.png |
| What’s the difference among println(), print() and printf()?  **print()**: print() method in Java is used to display a text on the console. This text is passed as the parameter to this method in the form of String. This method prints the text on the console  **println()**: println() method in Java is also used to display a text on the console. This text is passed as the parameter to this method in the form of String. This method prints the text on the console and the cursor remains at the start of the next line at the console.  The printf() method of Java PrintStream class is a convenience method to write a String which is formatted to this output Stream. It uses the specified format string and arguments. |
| Give the syntax for generic class  Syntax:  public class Box<T>  {  private T t;  }  Where  Box − Box is a generic class.  T − The generic type parameter passed to generic class. It can take any  Object.  t − Instance of generic type T. |
| Give two advantages of Generic programming.  Type-safety : We can hold only a single type of objects in generics. It  doesn’t allow to store other objects.  Type casting is not required: There is no need to typecast the object. |
| Compare expression lambdas and block lambdas? Give an example  **Lambda expression** is an unnamed method that is not executed on its own. These expressions cause anonymous class.  Lambdas which contain expression bodies are known as “Expression Lambdas“.  Block Lambda contains many operations that work on lambda expressions as it allows the lambda body to have many statements.  This includes variables, loops, conditional statements like if, else and switch statements, nested blocks, etc. This is created by enclosing the block of statements in lambda body within braces {}.  **Syntax:** Lambda Expressions  (parameters) -> { lambda body } |
| Define functional interface  An Interface that contains exactly one abstract method is known as functional interface. It can have any number of default, static methods but can contain only one abstract method. |
| What is mutable object and immutable object?  Mutable and immutable objects in Java, Mutable objects are those whose state can be changed after they have been created, while immutable objects are those whose state cannot be changed after they have been created. |
| What is the main difference between StringBuffer and StringBuilder?  String is immutable whereas StringBuffer and StringBuilder are mutable classes. StringBuffer is thread-safe and synchronized whereas StringBuilder is not. That's why StringBuilder is faster than StringBuffer. |
| What are the different ways of constructing a String object?  **There are two ways to create a String object:**   * By string literal : Java String literal is created by using double quotes. For Example: String s=“Welcome”; * By new keyword : Java String is created by using a keyword “new”. For example: String s=new String(“Welcome”); |
| What is the difference between equals() and == in Java Strings?  The major difference between the == operator and . equals() method is that one is an operator, and the other is the method. Both these == operators and equals() are used to compare objects to mark equality. |
| Write a java program to check whether a given string is palindrome or not.  import java.util.\*;  class Palindrome  {  public static void main(String args[])  {  String original, reverse = ""; // Objects of String class  Scanner in = new Scanner(System.in);  System.out.println("Enter a string/number to check if it is a palindrome");  original = in.nextLine();  int length = original.length();  for ( int i = length - 1; i >= 0; i-- )  reverse = reverse + original.charAt(i);  if (original.equals(reverse))  System.out.println("Entered string/number is a palindrome.");  else  System.out.println("Entered string/number isn't a palindrome.");  }  } |
| Differentiate between an ArrayList and an LinkedList  6136becaea75de0fea4ab6d9_arraylist-vs-linkedlist.jpg |
| What do you understand by Iterator in the Java Collection Framework?  Iterator in Java is an object used to cycle through arguments or elements present in a collection. It is derived from the technical term “iterating,” which means looping through. Generally, an iterator in Java is used to loop through any collection of objects. |
| What is the use of the List interface?  The List interface in Java provides a way to store the ordered collection. It is a child interface of Collection. It is an ordered collection of objects in which duplicate values can be stored. Since List preserves the insertion order, it allows positional access and insertion of elements. |
| What is a Vector in Java?  The Vector class implements a growable array of objects. Like an array, it contains components that can be accessed using an integer index. However, the size of a Vector can grow or shrink as needed to accommodate adding and removing items after the Vector has been created. |
| What is the HashSet class in Java and how does it store elements?  Java HashSet class is used to create a collection that uses a hash table for storage. It inherits the AbstractSet class and implements Set interface. HashSet stores the elements by using a mechanism called **hashing.** |
| What is LinkedHashSet in Java Collections Framework?  LinkedHashSet maintains a linked list of the entries in the set, in the order in which they were inserted. This allows insertion-order iteration over the set. That is, when cycling through a LinkedHashSet using an iterator, the elements will be returned in the order in which they were inserted. |
| What are the advantages of the Collection Framework in Java?  Java Collection framework has several advantages, few of them include: **1. Consistent API**  **2. Reduce Programming Exertion:**  **3. Increases Program Quality and Speed:** |
| What is Map interface in Java?  Map in Java is an interface available in java. util package and it stores the data in key and value pairs. It does not allow duplicate keys. |
| Why Map doesn’t extend the Collection Interface?  Map doesn’t extend the Collection Interface because Because they are of an incompatible type.  List, Set and Queue are a collection of similar kind of objects but just values where a Map is a collection of key and value pairs. |
| What are different ways to iterate over a list?  Simple For loop.  Enhanced For loop.  Iterator.  ListIterator.  While loop.  Iterable.forEach() util. |
| What are the basic interfaces of Java Collections Framework?  Collection — the root of the collection hierarchy. ...  Set — a collection that cannot contain duplicate elements. ...  List — an ordered collection (sometimes called a sequence). ...  Queue — a collection used to hold multiple elements prior to processing. |
| **Part – B ( 5 x 13 = 65 Marks)** |
| 11 a. Explain in detail the different states of a thread and thread priorities. (13) |
| 11 a. What is Thread? Explain Multithreading and Multitasking in detail. (13) |
| 11 a. Define Thread. Differentiate Multithreading and Multitasking. (13) |
| 11 a. Explain creation of a single thread and multiple threads using an example program. (13) |
| 11 a. Describe the different types of thread creation with example program. (13) |

|  |
| --- |
| 11 b. What is Synchronization? Explain the different types of Synchronization in java. (13) |
| 11 b. Explain the importance of Synchronization and its types in java. (13) |
| 11 b. Explain Interthread Communication in detail with Neat Sketch. (13) |
| 11 b. Define Synchronization and elaborate on the various types of Synchronization in Java. (13) |
| 11 b. Discuss the need for Synchronization. Explain any two ways used to achieve synchronization in multithreading with a example program. (13) |
| 12 a. Explain byte stream and character stream in detail with a example java program (13) |
| 12 a. Explain standard I/O streams which is automatically created with the console with example java program (13) |
| 12 a. Write a java program to illustrate reading input from the user in the command line environment (console) using Buffered Reader class and Scanner class. (13) |
| 12 a. Write a java program to illustrate reading input from the user in the command line environment (console) using Console class. List out advantages. (13) |
| 12 a. Write a java program to demonstrate various Scanner class methods to read primitive data. (13) |
| 12 b. Write a program to copy the contents of one file to another (13) |
| 12 b. Write a program to read from a file and display the content on the screen. (13) |
| 12 b. Write a program to read from a standard input device and display the content on the screen. (13) |
| 12 b. Write a program to demonstrate the usage of FileInputStream and FileOutputStream class. (13) |
| 12 b. Write a program to demonstrate the usage of FileReader and FileWriter class. (13) |

|  |
| --- |
| 13 a. Explain lambda expression with example. (13) |
| 13 a.Explain the implementation of functional interface with example. (13) |
| 13 a. How the lambda expression provides the implementation of an interface? Give suitable example. (13) |
| 13 a. With an example demonstrate the following: (13)   1. Lambda expression with no parameter 2. Lambda expression with single parameter |
| 13 a. Write the syntax and explain lambda expression with / without parameters.(13) |
| 13 b. Explain string handling functions in java with example. (13) |
| 13 b. List the string class methods with suitable example. (13) |
| 13 b. Demonstrate the string handling methods in java with example. (13) |
| 13 b. Determine the usage of string handling methods in java. (13) |
| 13 b. Explain with example for the methods provided by java.lang.String class. (13) |
| 14 a. Explain the various collection interfaces in java with example. (13) |
| 14 a. Describe the methods declared in collection interface with example. (13) |
| 14 a. Explain the hierarchy of collection framework. (13) |
| 14 a. Define an ArrayList and explain its methods. (13) |
| 14 a. Explain the List interface with suitable example. (13) |
| 14 b. Write a java program to find the kth smallest element in an ArrayList. (13)  **Input format**  First Integer represents N value.  Next N values represent array elements. Last value describes K. |

|  |
| --- |
| **Input**  5  11 13 15 12 14  2  **Output**  K'th smallest element is 12 |
| 14 b. Write a program to get the hall details and store in the ArrayList and search the hall and display it's position details. (13)  **Input Format**  The first line of input is an integer which corresponds to the number ‘n’ of halls.  The n lines of input are the string which corresponds to the hall name.  The last line of input is the string which corresponds to the hall name to be searched.  **Input**  3  DSP AIK SAI SAI  **Output**  SAI hall is found in the list at position 2 |
| 14 b. Write a java program to find the kth largest element in an ArrayList. (13)  **Input format**  First Integer represents N value.  Next N values represent array elements. Last value describes K.  **Input** |

|  |
| --- |
| 5  11 13 15 12 14  2  **Output**  K'th largest element is 14 |
| 14 b. Write a java program to find the maximum and minimum element in an ArrayList. (13)  **Input format**  First Integer represents N value.  Next N values represent array elements.  **Input**  5  11 13 15 12 14  **Output**  15 11 |
| 14 b. Write a program to get the student names and store in the ArrayList and search for a particular name and display it's position details. (13)  **Input Format**  The first line of input is an integer which corresponds to the number ‘n’ of names. The n lines of input are the string which corresponds to the student name.  The last line of input is the string which corresponds to the student name to be searched.  **Input**  3  DSP AIK SAI |

|  |
| --- |
| SAI  **Output**  SAI hall is found in the list at position 2 |
| 15 a. Write a generic program to bring out the purpose of bounded types and explain. (13) |
| 15 a. Write a generic program to read two data items of different data types and display it on screen. (13) |
| 15 a. Write a program to find the sum and average of numbers in a generic array. (13) |
| 15 a. Write a Generic method to print array of data which belongs to any class type. Analyse generics working with primitive types. (13) |
| 15 a. Write a program to find the average of numbers in a generic array. (13) |
| 15 b. Explain map interface with example. (13) |
| 15 b. Explain any two interfaces for implementing map in java. Give example. (13) |
| 15 b. With a neat diagram explain the map hierarchy with example. (13) |
| 15 b. Explain iterator in java with example. (13) |
| 15 b. Explain java iterator methods with suitable example. (13) |
| 16 a. Create a Bank database application program to illustrate the use of multithreads (15) |
| 16 a. Create an employee database application program to illustrate the use of multithreads (15) |
| 16 a. Create an Inventory application in Java to illustrate the use of multithreads (15) |
| 16 a. Write code to solve the produce consumer problem in Java (15) |
| 16 a. Write a program in java to Print even and odd numbers in increasing order using two threads (15) |
| 16 b. Write a program to obtain a string and identify whether it is a palindrome or not. (13)  **Input** madam  **Output**  madam is a Palindrome |

|  |
| --- |
| 16 b. Write a program to obtain a string, substring, and the string that has to replace the substring and print the modified string. (13)  **Input format**  The first line of the input consists of a string.  The second line of the input consists of a substring.  The third line of the input consists of a string that has to replace the substring.  **Input** audacious cio  aaa **Output** audaaaaus |
| 16 b. Write a program to accept a string and convert the first character of each word to uppercase. (13)  **Input**  india is my country  **Ouput**  India Is My Country |
| 16 b. Write a program to accept a string and count the number of vowels, consonants present in it. (13)  **Input** Hello world **Ouput** Vowels: 3  Consonants: 7 |
| 16 b. Write a program to accept a string and count the number of words present in it. (13)  **Input**  India is my country  **Ouput** |

|  |
| --- |
| 4 |