TCS NQT Questions and Answers 2024 - Coding, Reasoning, Aptitude

Preparing for the Tata Consultancy Services National Qualifier Test (TCS NQT) can be a challenging task. The TCS NQT questions vary widely from coding, aptitude, and reasoning, to verbal ability and more. This article serves as your comprehensive guide to effectively prepare for TCS NQT 2024, answering all your queries related to the TCS NQT exam questions, be it the advanced coding ones or numerical ability ones.

If you are looking for TCS NQT coding questions for 2024 or TCS NQT previous year questions, you've landed on the right page. We've collected different types of TCS NQT practice questions and provided detailed solutions to help you grasp the crux of the questions and their solutions. These include not just coding, but also TCS NQT aptitude questions, TCS NQT reasoning questions, and TCS NQT verbal ability questions.

Notably, we have also covered percentage, quantitative aptitude questions, and numerical ability questions for TCS NQT, making it a gold mine for aspirants aiming to crack the NQT with flying colours.

Finally, if you're seeking practice with English questions for TCS NQT or TCS NQT verbal questions, we've included those as well. Enhance your understanding, practice to perfection, and step confidently towards your dream job with TCS.

Download TCS NQT Previous Year Paper PDF here!

TCS NQT Coding Questions 2024

The coding section tests candidates on their programming acumen and logical reasoning skills. Questions range from data structures, algorithms, and string manipulations, to number series, and conversions. More emphasis has been laid on real-world problem-solving with code optimization and efficiency being a priority. Preparing for these coding questions requires a deep understanding of various programming concepts and rigorous practice. A strong grasp of one's chosen programming language is a must to solve the TCS NQT Coding Questions 2024 effectively and within the time limit.

Arrays Coding Questions

Question: Write a program to find the second largest number in an array.

Answer:

```
public class SecondLargestInArray {

public static void main(String[] args) {

int arr[] = {12, 35, 1, 10, 34, 1};

int largest = arr[0];

int secondLargest = arr[0];

for (int i = 0; i < arr.length; i++) {

if (arr[i] > largest) {

secondLargest = largest;

largest = arr[i];

} else if (arr[i] > secondLargest) {
```

```
secondLargest = arr[i];
}
}
System.out.println("The Second largest element in the array is: "+secondLargest);
}
}
```

Question: Write a program to reverse an array in-place.

Answer:

```
public class ReverseArray {
public static void main(String[] args) {
int arr[] = {10, 20, 30, 40, 50};
int start = 0;
int end = arr.length - 1;
while (start < end) {
int temp = arr[start];
arr[start] = arr[end];
arr[end] = temp;
start++;
end--;
}
System.out.println("Reversed array is: "+ Arrays.toString(arr));
}
}
```

Question: Write a program to find the duplicate values of an array of integer values.

Answer:

```
public class FindDuplicates {
public static void main(String[] args) {
int arr[] = {10, 20, 20, 30, 30, 30, 40, 50, 50};
System.out.println("Duplicate elements in array are: ");
for (int i = 0; i < arr.length; i++) {
for (int j = i + 1; j < arr.length; j++) {
if (arr[i] == arr[j]) {
```

```
System.out.println(arr[j]);
}
}
}
}
}
```

Question: Write a program to find the number of even and odd integers in a given array of integers.

Answer:

```
public class CountEvenOdd {
public static void main(String[] args) {
int arr[] = {10, 20, 21, 30, 31, 40, 41};
int even = 0;
int odd = 0;
for (int i = 0; i < arr.length; i++) {
if (arr[i] % 2 == 0) even++;
else odd++;
}
System.out.println("Number of even elements: "+even);
System.out.println("Number of odd elements: "+odd);
}
}
```

Question: Write a program to find the common elements between two arrays (string, integer, etc.) of sorted elements.

Answer:

```
public class CommonElements {
public static void main(String[] args) {
int[] arr1 = {1, 3, 4, 5, 7};
int[] arr2 = {2, 3, 5, 6};
System.out.println("Common elements are: ");
for(int i = 0; i < arr1.length; i++) {
for (int j = 0; j < arr2.length; j++) {
if(arr1[i] == arr2[j]) {
System.out.println(arr1[i]);
}

}

}

}

}
```

String Coding Questions

Question: Write a program to check whether a given string is a palindrome or not.

Answer:

```
public class Palindrome {

public static void main(String[] args) {

String str = "madam";

String reversed = new StringBuilder(str).reverse().toString();

if(str.equals(reversed)) {

System.out.println(str + " is a palindrome");

} else {

System.out.println(str + " is not a palindrome");

}

}

}
```

Question: Write a program to reverse words in a given sentence without using any library method.

Answer:

```
public class ReverseWords {

public static void main(String[] args) {

String str = "Hello World";

String[] words = str.split(" ");

String reversedString = "";

for (int i = 0; i < words.length; i++) {

String word = words[i];

String reversedWord = "";

for (int j = word.length()-1; j >= 0; j--) {

reversedWord = reversedWord + word.charAt(j);

}

reversedString = reversedString + reversedWord + " ";
}
System.out.println(reversedString);
}
}
```

Question: Write a program to count the total number of vowels and consonants in a string.

Answer:

```
public class CountVowelsConsonants {
public static void main(String[] args) {
String str = "Hello World";
int vowels = 0, consonants = 0;
for(int i = 0; i < str.length(); ++i) {
char ch = Character.toLowerCase(str.charAt(i));
if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
++vowels;
} else if ((ch >= 'a'&& ch <= 'z')) {
++consonants;
}
}
System.out.println("Number of vowels: " + vowels);
System.out.println("Number of consonants: " + consonants);
}
}
```

Question: Write a program to check whether two strings are anagrams of each other.

Answer:

```
public class AnagramCheck {
public static void main(String[] args) {
String str1 = "Listen";
String str2 = "Silent";
str1 = str1.toLowerCase();
str2 = str2.toLowerCase();
if(str1.length() == str2.length()) {
char[] charArray1 = str1.toCharArray();
char[] charArray2 = str2.toCharArray();
Arrays.sort(charArray1);
Arrays.sort(charArray2);
boolean result = Arrays.equals(charArray1, charArray2);
if(result) {
System.out.println(str1 + " and " + str2 + " are anagrams.");
} else {
System.out.println(str1 + " and " + str2 + " are not anagrams.");
}
} else {
System.out.println(str1 + " and " + str2 + " are not anagrams.");
}
}
}
```

Question: Write a program to convert a given string into lowercase.

Answer:

```
public class LowercaseString {
public static void main(String[] args) {
String str = "HELLO WORLD";
str = str.toLowerCase();
System.out.println("String in lowercase: "+str);
}
}
```

Number Series Coding Questions

Question: Write a program to print Fibonacci series up to n.

Answer:

```
public class Fibonacci {
public static void main(String[] args) {
int n = 10;
int t1 = 0, t2 = 1;
System.out.print("First " + n + " terms: ");
for (int i = 1; i <= n; ++i) {
System.out.print(t1 + " ");
int sum = t1 + t2;
t1 = t2;
t2 = sum;
}
}
}
```

Question: Write a program to print all prime numbers up to n.

Answer:

```
public class PrimeNumbers {
public static void main(String[] args) {
int n = 30;
System.out.println("Prime numbers up to " + n);
for(int i = 2; i <= n; i++) {
if(isPrime(i)) {
System.out.print(i + " ");
}
}
}
public static boolean isPrime(int num) {
if(num <= 1) {
return false;
}
for(int i = 2; i < num; i++) {
if(num % i == 0) {
return false;
}
}
return true;
}
}
```

Question: Write a program to print the Armstrong numbers between 1 and n.

Answer:

```
public class ArmstrongNumbers {
public static void main(String[] args) {
int n = 500;
for(int i = 1; i <= n; i++) {
if(isArmstrong(i)) {
System.out.print(i + " ");
}
}
}
public static boolean isArmstrong(int num) {
int original = num;
int sum = 0;
while (num != 0) {
int digit = num % 10;
sum += Math.pow(digit, 3);
num /= 10;
}
if(sum == original) {
return true;
}
return false;
}
}
```

Question: Write a program to print the pattern series like 1, 12, 123, 1234, ..., up to n.

Answer:

```
public class PatternSeries {
public static void main(String[] args) {
int n = 5;
for (int i = 1; i <= n; i++) {
for (int j = 1; j <= i; j++) {
System.out.print(j);
}
System.out.println();

}

}

}
```

Question: Write a program to print the sum of the series 1 + 1/2 + 1/3 + ... + 1/n.

Answer:

```
public class SeriesSum {

public static void main(String[] args) {

int n = 5;

double result = 0.0;

for (int i = 1; i <= n; i++) {

result += 1.0 / i;

}

System.out.println("The sum of the series is " + result);

}

}
```

Conversions Coding Questions

Question: Write a program to convert a decimal number to binary.

Answer:

```
public class DecimalToBinary {

public static void main(String[] args) {

int num = 10;

String binary = Integer.toBinaryString(num);

System.out.println("Binary representation of " + num + " is " + binary);

}

}
```

Question: Write a program to convert a binary number to decimal.

Answer:

```
public class BinaryToDecimal {

public static void main(String[] args) {

String binary = "1010";

int decimal = Integer.parseInt(binary, 2);
System.out.println("Decimal representation of " + binary + " is " + decimal);
}
}
```

Question: Write a program to convert a decimal number to hexadecimal.

Answer:

```
public class DecimalToHexadecimal {
public static void main(String[] args) {
int num = 255;
String hex = Integer.toHexString(num);
System.out.println("Hexadecimal representation of " + num + " is " + hex);
}
}
```

Question: Write a program to convert Fahrenheit to Celsius.

Answer:

```
public class FahrenheitToCelsius {
public static void main(String[] args) {
double fahrenheit = 100.0;
double celsius = (fahrenheit - 32) * 5 / 9;
System.out.println(fahrenheit + " degree Fahrenheit is equal to " + celsius + " in Celsius");
}
}
```

Question: Write a program to convert a string to an integer without using Integer.parseInt().

Answer:

```
public class StringToInteger {
public static void main(String[] args) {
String str = "123";
int num = 0;
for (char c : str.toCharArray()) {
num = num * 10 + (c - '0');
}
System.out.println("The integer value of " + str + " is " + num);
}
}
```

Sorting Algorithm Coding Questions

Question: Implement the bubble sort algorithm for a given array of integers.

Answer:

```
public class BubbleSort {
public static void main(String[] args) {
int[] array = {64, 34, 25, 12, 22, 11, 90};
bubbleSort(array);
System.out.println("Sorted array: " + Arrays.toString(array));
}
public static void bubbleSort(int[] array) {
int n = array.length;
for (int i = 0; i < n-1; i++) {
for (int j = 0; j < n-i-1; j++) {
if (array[j] > array[j+1]) {
int temp = array[j];
array[j] = array[j+1];
array[j+1] = temp;
}
}
}
}
}
```

Question: Implement the insertion sort algorithm for a given array of integers.

Answer:

```
public class InsertionSort {
public static void main(String[] args) {
int[] array = {12, 11, 13, 5, 6};
insertionSort(array);
System.out.println("Sorted array: " + Arrays.toString(array));
}
public static void insertionSort(int[] array) {
```

```
int n = array.length;
for (int i = 1; i < n; ++i) {
int key = array[i];
int j = i - 1;
while (j >= 0 && array[j] > key) {
array[j + 1] = array[j];
j = j - 1;
}
array[j + 1] = key;
}
}
}
```

Question: Implement the selection sort algorithm for a given array of integers.

Answer:

```
public class SelectionSort {
public static void main(String[] args) {
int[] array = {64, 25, 12, 22, 11};
selectionSort(array);
System.out.println("Sorted array: " + Arrays.toString(array));
}
public static void selectionSort(int[] array) {
int n = array.length;
for (int i = 0; i < n - 1; i++) {
int minIndex = i;
for (int j = i + 1; j < n; j++) {
if (array[j] < array[minIndex]) {
minIndex = j;
}
}
int temp = array[minIndex];
array[minIndex] = array[i];
array[i] = temp;
}

}

}
```

Question: Implement the quick sort algorithm for a given array of integers.

Answer:

```
public class QuickSort {

public static void main(String[] args) {

int[] array = {10, 7, 8, 9, 1, 5};

int n = array.length;

quickSort(array, 0, n - 1);

System.out.println("Sorted array: " + Arrays.toString(array));

}

public static int partition(int[] array, int low, int high) {

int pivot = array[high];

int i = (low - 1);

for (int j = low; j < high; j++) {

if (array[j] < pivot) {

i++;

int temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

int temp = array[i + 1];

array[i + 1] = array[high];

array[high] = temp;

return (i + 1);

}

public static void quickSort(int[] array, int low, int high) {

if (low < high) {

int pi = partition(array, low, high);

quickSort(array, low, pi - 1);
quickSort(array, pi + 1, high);
}
}
}
```

Question: Implement the merge sort algorithm for a given array of integers.

Answer:

```
public class MergeSort {
public static void main(String[] args) {
int[] array = {12, 11, 13, 5, 6, 7};
int n = array.length;
mergeSort(array, 0, n - 1);
System.out.println("Sorted array: " + Arrays.toString(array));
}
public static void mergeSort(int[] array, int left, int right) {
if (left < right) {
int mid = (left + right) / 2;
mergeSort(array, left, mid);
mergeSort(array, mid + 1, right);
merge(array, left, mid, right);
}
}
public static void merge(int[] array, int left, int mid, int right) {
int n1 = mid - left + 1;
int n2 = right - mid;
int[] Left = new int[n1];
int[] Right = new int[n2];
for (int i = 0; i < n1; ++i)
Left[i] = array[left + i];
for (int j = 0; j < n2; ++j)
Right[j] = array[mid + 1 + j];
int i = 0, j = 0;
int k = left;
while (i < n1 && j < n2) {
if (Left[i] <= Right[j]) {
array[k] = Left[i];
i++;
} else {
array[k] = Right[j];
j++;
}
k++;
}
while (i < n1) {
array[k] = Left[i];
i++;
k++;
}
while (j < n2) {
array[k] = Right[j];
j++;
k++;
}
}
}
```

This completes the merge sort algorithm where the array is first divided into two halves and then they are sorted and merged. The merge operation is crucial which operates on sorted arrays and merges them to form a larger sorted array.

TCS NQT Advanced Coding Questions

Question: Write a program to solve the Towers of Hanoi problem with 'n' disks.

Answer:

```
public class TowersOfHanoi {
public static void main(String[] args) {
int n = 3;
towerOfHanoi(n, 'A', 'C', 'B');
}
static void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod) {
if (n == 1) {
System.out.println("Move disk 1 from rod " + from_rod + " to rod " + to_rod);
return;
}
towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
System.out.println("Move disk " + n + " from rod " + from_rod + " to rod " + to_rod);
towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
}
}
```

Question: Write a program to find the Longest Common Subsequence (LCS) in two strings.

Answer:

```
public class LCS {
public static void main(String[] args) {
String s1 = "AGGTAB";
String s2 = "GXTXAYB";
int result = lcs(s1, s2, s1.length(), s2.length());
System.out.println("Length of LCS is " + result);
}
static int lcs(String s1, String s2, int m, int n) {
if (m == 0 || n == 0)
return 0;
if (s1.charAt(m - 1) == s2.charAt(n - 1))
return 1 + lcs(s1, s2, m - 1, n - 1);
else
return Math.max(lcs(s1, s2, m, n - 1), lcs(s1, s2, m - 1, n));
}
}
```

Question: Write a program to find all permutations of a string.

Answer:

```
public class StringPermutation {
public static void main(String[] args) {
```

```
String str = "ABC";
int n = str.length();
permute(str, 0, n - 1);
}
private static void permute(String str, int l, int r) {
if (l == r)
System.out.println(str);
else {
for (int i = l; i <= r; i++) {
str = swap(str, l, i);
permute(str, l + 1, r);
str = swap(str, l, i);
}
}
}
public static String swap(String a, int i, int j) {
char temp;
char[] charArray = a.toCharArray();
temp = charArray[i];
charArray[i] = charArray[j];
charArray[j] = temp;
return String.valueOf(charArray);
}
}
```

These questions are designed to test your problem-solving abilities and your familiarity with advanced algorithms and data structures. Each problem requires a different strategy, and the best way to prepare is to practice a variety of problems and understand the underlying algorithms and techniques.

Check TCS NQT Syllabus and Exam Pattern Now!

TCS NQT Programming Logic Questions 2024

The Tata Consultancy Services (TCS) National Qualifier Test (NQT) 2024 is expected to challenge the problem-solving skills of aspiring software professionals with a comprehensive set of programming logic questions. These questions aim to assess the ability to devise effective solutions for complex problems, as well as proficiency in various programming languages. Whether it's data structures, algorithms, control structures, or other foundational elements, these TCS NQT programming logic questions aim to test the depth of understanding and practical application. Thorough preparation and regular practice are pivotal in acing this segment, providing a competitive edge in the increasingly demanding landscape of tech recruitment.

Data Structures and Algorithms Questions

Question: Write an algorithm to implement the breadth-first search (BFS) for a graph.

Answer:

```
def BFS(graph, start):
visited = [] # List to keep track of visited nodes.
queue = [] # Initialize a queue
visited.append(start)
queue.append(start)
while queue:
m = queue.pop(0)
print (m, end = " ")
for neighbour in graph[m]:
if neighbour not in visited:
visited.append(neighbour)
queue.append(neighbour)
```

Question: Write an algorithm to implement the depth-first search (DFS) for a graph.

Answer:

```
def DFS(graph, start, visited=None):
if visited is None:
visited = set()
visited.add(start)
print(start, end=' ')
for next_node in graph[start] - visited:
DFS(graph, next_node, visited)
return visited
```

Question: Implement an algorithm to sort an array using a heap sort method.

Answer:

```
def heapify(arr, n, i):
largest = i # Initialize largest as root
l = 2 * i + 1
r = 2 * i + 2
if l < n and arr[i] < arr[l]:
```

```
largest = l
if r < n and arr[largest] < arr[r]:
largest = r
if largest != i:
arr[i],arr[largest] = arr[largest],arr[i]
heapify(arr, n, largest)
def heapSort(arr):
n = len(arr)
for i in range(n//2 - 1, -1, -1):
heapify(arr, n, i)
for i in range(n-1, 0, -1):
arr[i], arr[0] = arr[0], arr[i]
heapify(arr, i, 0)
```

Question: Write a program to find the shortest path between two nodes in a weighted graph using Dijkstra's Algorithm.

Answer:

```
import sys
def dijkstra(graph, start_vertex):
D = {v:float('infinity') for v in graph}
D[start_vertex] = 0
unseen_nodes = list(graph)
while unseen_nodes:
min_node = None
for node in unseen_nodes:
if min_node is None:
min_node = node
elif D[node] < D[min_node]:
min_node = node
for neighbour, weight in graph[min_node].items():
if weight + D[min_node] < D[neighbour]:
D[neighbour] = weight + D[min_node]
unseen_nodes.remove(min_node)
return D
```

Question: Write a program to find the InOrder, PreOrder and PostOrder traversal of a Binary Tree

Question: Write a program to find the InOrder, PreOrder and PostOrder traversal of a Binary Tree.

Answer:

```
class Node:
def __init__(self, key):
self.left = None
self.right = None
self.val = key
def printInorder(root):
if root:
printInorder(root.left)
print(root.val),
printInorder(root.right)
def printPostorder(root):
if root:
printPostorder(root.left)
printPostorder(root.right)
print(root.val),
def printPreorder(root):
if root:
print(root.val),
printPreorder(root.left)
printPreorder(root.right)
```

OOPS Concepts Questions

Question: Explain the concept of encapsulation with a coding example.

Answer: Encapsulation is one of the four fundamental OOP concepts. The other three are inheritance, polymorphism, and abstraction. Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class are hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

Here is a simple example of encapsulation that includes encapsulation, which contains two parts: Data hiding + getter and setter methods.

```
public class Employee {
private String name;
public String getName() {
return name;
}
public void setName(String newName) {
name = newName;
}
}
```

Question: Write a program to implement polymorphism in Java.

Answer: Polymorphism in Java is a concept by which we can perform a single action in different ways. Polymorphism allows us to perform a single action in many ways. In other words, polymorphism allows you to define one interface and have multiple implementations. Here is a simple example:

```
class Animal {
public void animalSound() {
System.out.println("The animal makes a sound");
}
}
class Pig extends Animal {
public void animalSound() {
System.out.println("The pig says: wee wee");
}
}
class Dog extends Animal {
public void animalSound() {
System.out.println("The dog says: bow wow");
}
}
public class Main {
public static void main(String[] args) {
Animal myAnimal = new Animal();
Animal myPig = new Pig();
Animal myDog = new Dog();
myAnimal.animalSound();
myPig.animalSound();
myDog.animalSound();
}
}
```

Question: What is the difference between overloading and overriding? Provide a code example.

Answer: Overloading in Java is the ability to create multiple methods of the same name, but with different parameters. Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by its parent class or superclass. Overriding is done using inheritance.

Example of Overloading:

```
public class DemoClass {

void demoMethod(int i) {

System.out.println("Method with int: " + i);

}

void demoMethod(double d) {

System.out.println("Method with double: " + d);

}

}
```

Example of Overriding:

java

Copy code

```
class Animal {

void eat() {

System.out.println("eating...");

}

}

class Dog extends Animal {

void eat() {

System.out.println("eating bread...");

}

}
```

Question: Explain the concept of inheritance with an example. What are the different types of inheritance supported in Java and C++?

Answer: Inheritance is one of the key features of Object Oriented Programming. It allows a Class to inherit the properties (data members) and methods (functions) of another class. In terms of inheritance, classes are categorized into two types - parent (or base) class and child (or derived) class.

Example:

```
class Bicycle {

public int gear;

public int speed;
public int speed;
public Bicycle(int gear, int speed) {
this.gear = gear;
this.speed = speed;
}
public void applyBrake(int decrement) {
speed -= decrement;
}
public void speedUp(int increment) {
speed += increment;
}
}
class MountainBike extends Bicycle {
public int seatHeight;
public MountainBike(int gear,int speed,int startHeight) {
super(gear, speed);
seatHeight = startHeight;
}
public void setHeight(int newValue) {
seatHeight = newValue;
}
}
```

Java supports Single Inheritance and Multilevel Inheritance but doesn't support Multiple Inheritance directly. It can be achieved through interfaces though.

C++ supports all types of inheritance: Single, Multiple, Multilevel, Hierarchical and Hybrid.

Question: Write a program to implement an interface in Java. Also explain the difference between a class and an interface.

Answer: An interface in Java is a blueprint of a class. It has static constants and abstract methods only. The interface in Java is a mechanism to achieve fully abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve fully abstraction and multiple inheritance in Java.

Example:

```
interface Animal {
public void eat();
public void sound();
}
class Cat implements Animal {
```

```
class Cat implements Animal {
public void eat() {
System.out.println("Cat eats");
}
public void sound() {
System.out.println("Cat meows");
}
}
public class Main {
public static void main(String[] args) {
Cat c1 = new Cat();
c1.eat();
c1.sound();
}
}
```

The main differences between a Java class and a Java interface are:

- A class can implement several interfaces but can only extend one class.
- A class can have constructors and destructors, but an interface cannot.
- A class can have static methods, but an interface can't.
- A class can have all types of data members (private, protected, public), but an interface can only have public static final variables (constants).
- An interface provides a form of multiple inheritance. A class can extend only one other class.

Check TCS NQT Preparation Tips Now!

TCS NQT Quantitative Aptitude Questions 2024

Question 1: A factory produces x units of a product in 24 days. How many units will it produce in 15 days?

Answer:

Let's assume the rate of production is R units per day.

As per the question, R = x/24 (units per day)

To find the units produced in 15 days, we can use the formula:

Units produced in 15 days = R * 15 = (x/24) * 15 = (15x)/24 = 5x/8

Therefore, the factory will produce 5x/8 units in 15 days.

Question 2: A man buys a shirt at a discount of 25% on its marked price and sells it at a profit of 20% on its marked price. If the marked price is $400, what is the selling price of the shirt?

Answer:

Marked Price = $400

Discount = 25% of $400 = (25/100) * 400 = $100

Selling Price = Marked Price - Discount = $400 - $100 = $300

Profit = 20% of $400 = (20/100) * 400 = $80

Selling Price = Selling Price + Profit = $300 + $80 = $380

Therefore, the selling price of the shirt is $380.

Question 3: If a cube has a volume of 512 cubic centimeters, what is the length of its side?

Answer:

The volume of a cube is given by the formula: Volume = side * side * side

Given that the volume is 512 cubic centimeters, we have:

512 = side * side * side

Taking the cube root of both sides:

side = $\sqrt[3]{(512)} = 8$

Therefore, the length of one side of the cube is 8 centimeters.

Question 4: If a car travels a distance of 240 km with a speed of 60 km/h for the first 4 hours and then with a speed of 40 km/h for the next 3 hours, what is the average speed of the car for the entire journey?

Answer:

Total distance traveled = 240 km

Total time taken = 4 hours + 3 hours = 7 hours

Average speed = Total distance / Total time

Average speed = 240 km / 7 hours ≈ 34.29 km/h (rounded to two decimal places)

Therefore, the average speed of the car for the entire journey is approximately 34.29 km/h.

Question 5: If the cost price of an item is $200 and the selling price is $250, what is the profit percentage?

Answer:

Profit = Selling Price - Cost Price

Profit = $250 - $200 = $50

Profit Percentage = (Profit / Cost Price) * 100

Profit Percentage = (50 / 200) * 100 = 25%

Therefore, the profit percentage is 25%.

TCS NQT Verbal Ability Questions 2024

Question 1: Choose the correct synonym for the word "Ephemeral":

1. a) Permanent
2. b) Fleeting
3. c) Eternal
4. d) Prolonged

Answer: b) Fleeting

Answer: b) Fleeting

Question 2: Identify the correctly spelled word:

1. a) Exaggerrate
2. b) Exagerate
3. c) Exaggerate
4. d) Exaggarate

Answer: c) Exaggerate

Question 3: Choose the sentence with the correct subject-verb agreement:

1. a) The group of students has started their project.
2. b) The group of students have started their project.
3. c) The group of students is starting their project.
4. d) The group of students are starting their project.

Answer: c) The group of students is starting their project.

Question 4: Choose the appropriate word to complete the sentence:

He was ______ tired that he fell asleep immediately.

1. a) Too
2. b) Two
3. c) To
4. d) 2

Answer: a) Too

Question 5: Identify the correct sentence with the use of articles:

1. a) I have a apple.
2. b) I have an apple.
3. c) I have the apple.
4. d) I have apple.

Answer: b) I have an apple.

TCS NQT Numerical Ability Questions 2024

Question 1: Simplify the following expression:

(8 * 4 + 6) / 2 - 5

Answer:

(8 * 4 + 6) / 2 - 5

= (32 + 6) / 2 - 5

= 38 / 2 - 5

= 19 - 5

= 14

Question 2: Find the value of 'x' in the equation:

3x + 10 = 31

Answer:

$3x + 10 = 31$

$3x = 31 - 10$

$3x = 21$

$x = 21 / 3$

$x = 7$

Question 3: A bookstore gives a 15% discount on all books. If the original price of a book is $40, what will be the discounted price?

Answer:

Discounted price = Original price - (Original price * Discount percentage)

Discounted price = 40 - (40 * 0.15)

Discounted price = 40 - 6

Discounted price = $34

Question 4: The sum of two consecutive odd numbers is 64. What are the numbers?

Answer:

Let the first odd number be x.

The next consecutive odd number will be (x + 2).

According to the given condition:

$x + (x + 2) = 64$

$2x + 2 = 64$

$2x = 64 - 2$

$2x = 62$

$x = 62 / 2$

$x = 31$

So, the two consecutive odd numbers are 31 and 33.

Question 5: A train travels a distance of 240 km at a constant speed. If the speed of the train is 60 km/h, how much time will it take to cover the distance?

Answer:

Time taken = Distance / Speed

Time taken = 240 km / 60 km/h

Time taken = 4 hours

Therefore, it will take 4 hours for the train to cover the distance of 240 km.

TCS NQT Reasoning Questions 2024

Question 1: Choose the next number in the series:

2, 5, 11, 20, ?

2, 5, 11, 20, ?

Answer:

The series follows the pattern of adding consecutive odd numbers to the previous number: 2+3=5, 5+6=11, 11+9=20.

So, the next number will be 20+12=32.

Therefore, the next number in the series is 32.

Question 2: If APPLE is coded as 98553, how is ORANGE coded?

Answer:

Each letter in APPLE is replaced by its corresponding position in the English alphabet:

A=1, P=16, L=12, E=5.

So, APPLE is coded as 98553.

Using the same coding scheme, ORANGE will be coded as:

O=15, R=18, A=1, N=14, G=7, E=5.

Therefore, ORANGE is coded as 151817145.

Question 3: Which word does not belong to the group?

1. a) Square
2. b) Circle
3. c) Rectangle
4. d) Triangle

Answer:

1. b) Circle

All the other options are geometric shapes with straight lines, while a circle is a curved shape.

Question 4: If CAT is coded as 312, DOG is coded as 415, how is FISH coded?

Answer:

Each letter in the word is replaced by its corresponding position in the English alphabet:

C=3, A=1, T=20.

So, CAT is coded as 312.

Using the same coding scheme, FISH will be coded as:

F=6, I=9, S=19, H=8.

Therefore, FISH is coded as 6919198.

Question 5: Arrange the following words in alphabetical order:

Monitor

Apple

Keyboard

Mouse

Answer:

The words in alphabetical order are:

Apple

Keyboard

Monitor

Mouse

We hope you have liked the article on TCS NQT questions. Download the Testbook App for free and start your preparation for the TCS NQT Exam 2024 with expert guidance.