Problem Solving, Mathematical Thinking & Algorithms (60 Minutes) [17-Feb-2020]

1. Which of the following thing to keep in mind while solving a problem is?
   1. Input Data
   2. Output Data
   3. Stored Data
   4. All of the above
2. Which of the following is first step in process of problem solving?
   1. Design a Solution
   2. Define a problem
   3. Practicing the solution
   4. Organizing the data
3. ___________________ is creative technique in which you arrive at creative ideas by jotting them down yourself rather than working in a group.
   1. Brain storming
   2. Brain mining
   3. Brain writing
   4. Brain Reading
4. A major challenge in developing creative thinking skills is to learn how to think _____________ in addition to ______________.
   1. Emotionally, rationally
   2. Laterally, vertically
   3. Logically, laterally
   4. Vertically, rationally
5. Which of the following doesn't help to improve creativity?
   1. Stick to the traditional mental set
   2. Discipline yourself to think laterally
   3. Conduct brainstorming session
   4. Concentrate intensely on the task at hand
6. Which of the following is not a characteristics of creative people?
   1. Can work along in isolation necessary for developing ideas
   2. A positive self-image without being blindly self-confident
   3. Frequently considered to be nonconformists and do not need strong approval from the group
   4. Have a low tolerance for ambiguity, must have clear directions on all task
7. The essence of decision-making is ________________________.
   1. Problem solving
   2. Choosing between alternatives
   3. Developing alternative courses of action
   4. Monitoring
8. Which of the following is NOT a heuristic associated with bounded rationality decision making?
   1. Procrastination
   2. Availability
   3. Representativeness
   4. Anchoring/judgement
9. _______________ is a common statistical method where assumptions are made to explain the problem at hand.
   1. Lateral Thinking

2. Hypothesis Testing
3. Means End Analysis
4. Analogy

10. _______________ is thinking out of the box. The major barrier to this strategy is the mindset.
    1. Lateral Thinking
    2. Hypothesis Testing
    3. Means End Analysis
    4. Analogy
11. A problem solving technique in which we need to find multiple solution for a given problem and choose the one which worked best is called?
    1. Means End Analysis
    2. Reduction
    3. Trial and Error
    4. Hypothesis Testing
12. Which of the following helps you plan on how to use your most feasible solution to solving a problem?
    1. You will need to verify whether the steps in the solution are being addressed
    2. You will need to schedule your entire approach. You must identify the time that you will take to solve your problem
    3. Both 1 and 2
    4. None of the above
13. The meaning of the word ___________________ is pretty clear cut. It stands for the in-depth analysis of a problem in order to arrive at the best possible solution in order to solve it.
    1. Critical Thinking
    2. Problem Solving
    3. Alternate Solutions
    4. Problem Breakdown
14. Which of the following is/are the elements that are part of the critical thinking process?
    1. Arguments
    2. Safeguarding the facts
    3. Review and Revise
    4. Both 1 and 2
15. When faced with a problem or a decision-making opportunity, you must simultaneously look for a ________________ within limits. It will help you concentrate better on the problem.
    1. Alternate Solution
    2. Distraction
    3. Deep Focus
    4. Dedication
16. Match the following correctly:

	Thinking Hats		**Role**
1	White Hat	A	Judgment
2	Black Hat	B	Feelings and Intuition
3	Red Hat	C	Creativity
4	Green Hat	D	Information Known

1. 1-C, 2-D, 3-A, 4-B
2. 1-A, 2-B, 3-C, 4-D
3. 1-B, 2-C, 3-D, 4-A
4. 1-D, 2-A, 3-B, 4-C

17. ______________ is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output.
    1. Algorithm
    2. Program
    3. Process
    4. Thread
18. The running time of quick sort depends heavily on the selection of ________________.
    1. Number of inputs
    2. Pivot Element
    3. Size of elements
    4. Arrangement of elements in the array
19. What is the full form of FMEA technique?
    1. Failure Mode and Effects Analysis
    2. Feature Mode and Effects Analysis
    3. Failure Mode and End Analysis
    4. Find Mean Expected Analysis
20. Finding the location of the element with the given value is _______________.
    1. Search
    2. Traversal
    3. Sort
    4. Mapping
21. Which of the following is an example of linear data structure?
    1. Trees
    2. Arrays
    3. Graph
    4. None of the above
22. Which of the following is an example of Non-Linear Data Structure?
    1. Arrays
    2. Linked List
    3. Stack
    4. Graph
23. Which sorting algorithm works by repeatedly exchanging adjacent elements, if necessary?
    1. Radix Sort
    2. Bubble Sort
    3. Selection Sort
    4. Insertion Sort
24. ______________ is used to solve the optimization problem. An optimization problem is one in which we are given a set of input values, which are required either to be maximized or minimized (known as objective), i.e. some constraints or conditions.
    1. Greedy Technique
    2. Divide and Conquer Approach
    3. Branch and Bound
    4. Backtracking Algorithm
25. Which of the following tries each possibility until they find the right one & it is a depth-first search of the set of possible solution.
    1. Greedy Technique
    2. Divide and Conquer Approach
    3. Branch and Bound

    4. Backtracking Algorithm

26. Determine the Time Complexity of the code provided below.

```
int count = 0;
for(int i=0; i<N; i++){
            for(int j=0; j<i; j++){
                        Count++;
            }
}
```

    1. O(N)
    2. O(N+i)
    3. O(N^2)
    4. O(N/i)

27. Linear Search Algorithm has Best time complexity of _____________ average time complexity of _____________ and worst time complexity of __________________.
    1. Omega (1), Theta (n), O(nlog(n))
    2. Omega(n), Theta(n), O(n log(n))
    3. Omega(1), Theta(n), O(n)
    4. Omega(1), Theta(n log(n)), O(n^2)

28. Insertion Sort algorithm has Best time complexity of ______________ average time complexity of _______________ and worst time complexity of _________________.
    1. Omega(1), Theta(n), O(n log(n))
    2. Omega(n), Theta(n^2), O(n^2)
    3. Omega(1), Theta(n), O(n)
    4. Omega(1), Theta(n log(n)), O(n^2)

29. Merge Sort algorithm has Best time complexity of ____________ average time complexity of __________ and worst time complexity of _________________.
    1. Omega(n), Theta(n log(n)), O(n log(n))
    2. Omega(n), Theta(n), O(n^2)
    3. Omega(1), Theta(n log(n)), O(n^2)
    4. Omega(n log(n)), Theta(n log(n)), O(n log(n))

30. The time complexity of Fibonacci series is _____________________.
    1. O(log n)
    2. O(n(log n))
    3. O(n^2)
    4. O(2^n)

31. In which type of proof we assume that statement to be proved is false?
    1. Proof by Contradiction
    2. Proof by Induction
    3. Direct Proof
    4. Indirect Proof

32. If for an algorithm time complexity is given by O ((3/2) n) then complexity will be ___________.
    1. Constant
    2. Quadratic
    3. Exponential
    4. None of the above

33. What is the base of an undecimal system?
    1. 10
    2. 11

3. 12
4. 6

34. What is the decimal expansion of $(2AE0B)_{16}$?
    1. $(175627)_{10}$
    2. $(175883)_{10}$
    3. $(178586)_{10}$
    4. $(175562)_{10}$
35. By using the Euclidean algorithm, the linear combination of gcd(252, 198) = 18 is ___________.
    1. 252*4 – 198*5
    2. 252*5 – 198*4
    3. 252*5 – 198*2
    4. 252*4 – 198*4
36. ______________ is the measurement of memory space requirements of an algorithm.
    1. Time Complexity
    2. Space Complexity
    3. Best-case Complexity
    4. Worst-case Complexity
37. The time complexity to perform the modular exponentiation of a = cg (mod m) is ___________.
    1. O(m + a)
    2. O(a*g)
    3. O(gm)
    4. O(g)
38. Which component in Greedy Algorithms is used to determine whether a candidate can be used to determine whether a candidate can be used to contribute to the solution?
    1. A Feasibility Function
    2. A Selection Function
    3. A Candidate Set
    4. A Solution Function
39. ______________ always provide fixed predictable results for a given input.
    1. Exact Algorithms
    2. Non-Deterministic Algorithms
    3. Deterministic Algorithms
    4. Recursive Algorithms
40. Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?

    f1(n) = 2^n
    f2(n) = n^(3/2)
    f3(n) = nLog(n)
    f4(n) = n^(Logn)

    1. f3 < f2 < f4 < f1
    2. f3 < f2 < f1 < f4
    3. f2 < f3 < f1 < f4
    4. f2 < f3 < f4 < f1

OOPS & Data Structures using Java (60 Minutes)

1. In Threaded Binary Tree, left null link of node is replaced with address of its ______________.

   1. Inorder Predecessor
   2. Inorder Successor
   3. Preorder Successor
   4. Postorder Predecessor

2. What is return type of getParameterList() function of class javax.mail.internet.ContentType?
   1. Return the specified parameter value
   2. Return the MIME type string, without the parameters
   3. Return a ParameterList object that holds all the available parameters
   4. Return the subtype
3. Java NIO is ______________ oriented?
   1. IO
   2. Stream
   3. Buffer
   4. Process
4. What will s2 contain after following lines of Java code snippet?

   ```
   StringBuffer s1 = “one”;
   StringBuffer s2 = s1.append(“two”);
   ```

   1. onetwo
   2. Two
   3. One
   4. Compiler error
5. What is output of following code?

   ```
   class Outer
   {
       void outerMethod(){
           int x = 98;
           System.out.println(“inside outerMethod”);
           class Inner{
               void innerMethod(){
                   System.out.println(“x=”+x);
               }
           }
           Inner y = new Inner();
           y.innerMethod();
       }
   }
   class MethodLocalVariableDemo{
       public static void main(String[] args){
           Outer x = new Outer();
           x.outerMethod();
       }
   }
   ```

   1. inside outerMethod
      x = null
   2. x = 98
      inside outerMethod
   3. Compiler error
   4. Inside outerMethod

x = 98

6. What is output of following code?

```
public class Test implements Runnable
{
        public void run()
        {
                System.out.println(“GFG”);
        }
        public static void main(String []args) throws InterruptedException
        {
                Thread thread1 = new Thread(new Test());
                thread1.start();
                thread1.start();
                System.out.println(thread1.getState());
        }
}
```

   1. GFG GFG TERMINATED
   2. Runtime Error
   3. GFG TERMINATED
   4. Compilation Error

7.

```
public class MyClass{
        public static void main(String[] args){
                try{
                        int[] myNumbers = {1, 2, 3};
                        System.out.println(myNumbers[10]);
                }catch(Exception e){
                        System.out.println(“Something went wrong.”);
                        throw new ArithmeticException();
                }finally{
                        System.out.println(“The ‘try catch’ is finished.”);
                }
        }
}
```

   1. Something went wrong.
      The ‘try catch’ is finished.
      Exception in thread “main”
      java.lang.ArithmeticException
      at MyClass.main(MyClass.java:8)
   2. Something went wrong.
      The ‘try catch’ is finished.
   3. Exception in thread “main”
      Java.lang.ArithmeticException
      at MyClass.main(MyClass.java:8)
   4. Runtime error

8. Which function is used to perform some action when the object is to be destroyed?
   1. finalize()
   2. delete()
   3. main()

    4. destroy()

9. What will be the output for the following program?

```
class empty{
        private int num;
        public empty(){num=0;}
}
public class HelloWorld{
        public static void main(String []args){
                empty e1 = new empty();
                System.out.println(“num:”+e1.num);
        }
}
```

    1. 0
    2. Compilation Error
    3. Exception in thread “main” java.lang.StackOverflowError
    4. Exception in thread “main” java.lang.NullPointerException

10. Java compiler compiles Java Source Code into _______________.
    1. Bytecode
    2. Source Code
    3. .Obj
    4. .Exe

11. What will be time complexity of running merge sort on an array of size n which is already sorted?
    1. O(n)
    2. O(nlogn)
    3. $O(n^2)$
    4. $O(n^3)$

12. What will be the output of the code?

```
public class Main{
        public static void main(String[] args){
                String s1 = “Hello”;
                String s2 = “Hello”;
                String s3 = new String(s2);
                System.out.println(“”+(s1==s2));
        }
}
```

    1. False
    2. 0
    3. True
    4. 1

13. A sorting technique is called stable if it __________________.
    1. Takes O(nlogn) times
    2. Maintains the relative order of occurrence of non-distinct elements
    3. Uses divide-and-conquer paradigm
    4. Takes O(n) space

14. Breadth First Search is equivalent to which of the following traversal in the Binary Trees?

   1. Pre-order Traversal
   2. Post-order Traversal
   3. Level-order Traversal
   4. In-order Traversal

15. Which of the following algorithms can be used to most efficiently determine the presence of a cycle in a given graph?
    1. Depth First Search
    2. Breadth First Search
    3. Prim's Minimum Spanning Tree Algorithm
    4. Kruskal's Minimum Spanning Tree Algorithm
16. Which of the following is not a java keyword?
    1. Instanceof
    2. Sizeof
    3. Null
    4. volatile
17. More than one method with a same name and different arguments is called ______________.
    1. Operator Overloading
    2. Friend Function
    3. Method Overloading
    4. Method Overriding
18. By using which of the following technique the inheritance of a Class can be restricted?
    1. Parameterized constructor
    2. The final keyword
    3. Private constructors
    4. Both 2 and 3
19. Which of the following is the superclass of all exception classes?
    1. Throwable
    2. Exception
    3. RunTimeException
    4. IOException
20. What does the following function do for a given LinkedList with first node as head?

    ```
    void fun1(Node head)
    {
            if(head == NULL)
                    Return;
            fun1(head.next);
            System.out.println(head.data);
    }
    ```

    1. Prints all nodes of linked lists
    2. Prints all nodes of linked list in reverse order
    3. Prints alternate nodes of Linked List
    4. Prints alternate nodes in reverse order
21. Consider an implementation of unsorted circular linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O (1) time?
    1. Insertion at the front of the linked list
    2. Insertion at the end of the linked list
    3. Deletion of the front node of the linked list

    4. Deletion of the end node of the linked list

22. The concatenation of two list can performed in O (1) time. Which of the following variation of linked list can be used?
    1. Singly Linked List
    2. Doubly Linked List
    3. Circular Doubly Linked List
    4. Array Implementation of List
23. A variation of linked list is circular linked list, in which the last node in the list points to first node of the list. One problem with this type of list is?
    1. It waste memory space since the pointer head already points to the first node and thus the list node does not need to point to the first node.
    2. It is not possible to add a node at the end of the list
    3. It is difficult to traverse the list as the pointer of the last node is now not NULL
    4. All of the above
24. Which of the following code snippet will perform recursive Binary search, where recursiveBinarySearch() method accepts a sortedArray, key, the low and high indexes of the sortedArray.
    1. public int recursiveBinarySearch(int[] sortedArray, int key, int low, int high){
       ```
           int middle = (low+high)/2;
           if (high < low){return -1;}
           if (key == sortedArray[middle]){return middle;}
           else if (key < sortedArray[middle+1]){
                   return recursiveBinarySearch(sortedArray, key, low, middle - 1);
           else {
                   return recursiveBinarySearch(sortedArray, key, middle+1, high);
           }
       }
       ```
    2. public int recursiveBinarySearch(int[] sortedArray, int key, int low, int high){
       ```
           int middle = (low+high)/2;
           if (high < low){return -1;}
           if (key == sortedArray[middle]){return middle;}
           else if (key < sortedArray[middle]){
                   return recursiveBinarySearch(sortedArray, key, low, middle);
           else {
                   return recursiveBinarySearch(sortedArray, key, middle, high);
           }
       }
       ```
    3. public int recursiveBinarySearch(int[] sortedArray, int key, int low, int high){
       ```
           int middle = (low+high)/2;
           if (high < low){return -1;}
           if (key == sortedArray[middle]){return middle;}
           else if (key < sortedArray[middle]){
                   return recursiveBinarySearch(sortedArray, key, low, middle - 1);
           else {
                   return recursiveBinarySearch(sortedArray, key, middle+1, high);
           }
       }
       ```
    4. public int recursiveBinarySearch(int[] sortedArray, int key, int low, int high){

```
    int middle = (low+high)/4;
    if (high < low){return -1;}
    if (key == sortedArray[middle]){return middle;}
    else if (key < sortedArray[middle]){
        return recursiveBinarySearch(sortedArray, key, low, middle - 1);
    else {
        return recursiveBinarySearch(sortedArray, key, middle+1, high);
    }
  }
```

25. Consider the following code snippet of a Two Dimensional Array myNumbers, what would be the o/p of the following?

```
public class MyClass
{
    public static void main(String []args)
    {
        int[][] myNumbers = {{1, 2, 3, 4}, {5, 6, 7}};
        for(int i=0; i<myNumbers.length; ++i){
            for(int j=0; j<myNumbers.length; ++j){
                System.out.print(myNumbers[i][j]+” “);
            }
        }
    }
}
```

1. 1 2 3 4 5 6 7
2. 3 4 7
3. 2 3 4 6 7
4. 1 2 3 4 6 7

26. What will the output of the following code snippet, where Stack is a standard stack data structure?

- Pop() is an Utility function to remove top element from the stack
- Push() is an Utility function to add an element x in the stack
- Peek() is an Utility function to return top element in a stack but does not remove the element from the stack
- isEmpty() is an Utility function to check if the stack is empty or not, Returns true in case of Empty

```
public static void main(String []args)
{
    Stack stack = new Stack(3);
    stack.push(1);
    stack.push(2);
    stack.pop();
    stack.pop();

    stack.push(3);
    System.out.println(“Top element is: “+stack.peek());

    stack.pop();
    if(stack.isEmpty())
```

```
        System.out.println(“Stack Is Empty”);
    else
        System.out.println(“Stack Is Not Empty”);
}
```

1. Top element is: 3
   Stack Is Empty
2. Top element is: 2
   Stack Is Empty
3. Top element is: 2
   Stack Is Not Empty
4. Top element is: 3
   Stack Is Not Empty

27. What will the Output of the following code snippet, where Queue is a standard queue data structure?
    - enqueue() is an Utility function to add an item to the queue
    - dequeuer() is an Utility function to remove front element from the queue
    - peek() is an Utility function to return front element in the queue but does not remove it from the Queue

```
class Main
{
    public static void main(String[] args)
    {
        Queue q = new Queue(5);
        q.enqueue(1);
        q.enqueue(2);
        q.enqueue(3);

        System.out.println(“Front element is: “+q.peek());

        q.dequeue();
        q.dequeue();
        q.dequeue();

        if (q.isEmpty())
            System.out.println(“Queue is Empty”);
        else
            System.out.println(“Queue is Not Empty”);
    }
}
```

1. Front element is: 3
   Queue is Empty
2. Front element is: 1
   Queue Is Not Empty
3. Front element is: 1
   Queue Is Empty
4. Front element is: 2
   Queue Is Not Empty

28. What will be the output of below Snippet?

```
//Java program to demonstrate working of HashSet from Java Utils package
import java.util.*;

class Test
{
    public static void main(String []args)
    {
        HashSet<String> h = new HashSet<String>();

        h.add(“Pune”);
        h.add(“Mumbai”);
        h.add(“Bangalore”);
        h.add(“Mumbai”);

        System.out.println(“The List is “+h);
        System.out.println(“List contains Mumbai or not:”+h.contains(“Mumbai”));

        h.remove(“Australia”);
        System.out.println(“List after removing Australia:”+h);
    }
}
```

1. The List is [Pune, Mumbai, Banglore, Mumbai]
   List contains Mumbai or not: true
   List after removing Australia: [Pune, Mumbai, Banglore]
2. The List is [Pune, Mumbai, Banglore]
   List contains Mumbai or not: true
   List after removing Australia: [Pune, Mumbai, Banglore]
3. The List is [Pune, Mumbai, Banglore, Mumbai]
   List contains Mumbai or not: false
   List after removing Australia: [Pune, Mumbai, Banglore]
4. The List is [Pune, Mumbai, Banglore, Mumbai]
   List contains Mumbai or not: true
   List after removing Australia: [Pune, Mumbai, Banglore, Mumbai]

29. What will be the o/p of the below code snippet?

```
import java.util.*;

class TreeSetDemo{
    public static void main(String[] args)
    {
        TreeSet<String> ts1 = new TreeSet<String>();

        ts1.add(“Delhi”);
        ts1.add(“Mumbai”);
        ts1.add(“Bangalore”);
        ts1.add(“Mumbai”);

        System.out.println(ts1);
    }
}
```

1. [Mumbai, Mumbai, Delhi, Bangalore]
2. [Mumbai, Delhi, Bangalore]
3. [Bangalore, Delhi, Mumbai, Mumbai]
4. [Bangalore, Delhi, Mumbai]

30. What will be the o/p of the below code Snippet?

```
//Java program to demonstrate working of ArrayList in Java
import java.io.*;
import java.util.*;

class ArrayListTest{
        public static void main(String[] args) throws IOException
        {
                int size = 3;
                List<Integer> al = new ArrayList<>(size);

                for(int i=1; i<=size; i++)
                {
                        al.add(i);
                }

                for(int i=1; i<=size; i++)
                {
                        al.add(i);
                }
                al.remove(3);
                System.out.println(al);
        }
}
```

1. [1, 2, 3, 1, 2]
2. [1, 2, 3, 2, 3]
3. [1, 2, 3]
4. [1, 2, 3, 1, 3]

31. What will be the O/p of the below code Snippet?

```
import java.util.HashSet;
import java.util.Set;
import java.util.TreeSet;

public class testConversion{
        public static void main(String[] args)
        {
                Set<String> setobj = new HashSet<>();
                setobj.add("Pune");
                setobj.add("Mumbai");
                setobj.add("Goa");
                setobj.add("Bangalore");
                setobj.add("Chennai");
                System.out.println(“HashSet: “+setobj);
                Set<String> hashSetToTreeSet = new TreeSet<>(setobj);
                System.out.println(“TreeSet: “+hashSetToTreeSet);
        }
}
```

1. HashSet: [Chennai, Goa, Pune, Mumbai, Bangalore]
   TreeSet: [Bangalore, Chennai, Goa, Mumbai, Pune]
2. HashSet: [Bangalore, Chennai, Goa, Mumbai, Pune]
   TreeSet: [Bangalore, Chennai, Goa, Mumbai, Pune]
3. HashSet: [Chennai, Goa, Pune, Mumbai, Bangalore]
   TreeSet: [Chennai, Goa, Pune, Mumbai, Bangalore]
4. HashSet: [Bangalore, Chennai, Goa, Mumbai, Pune]
   TreeSet: [Chennai, Goa, Pune, Mumbai, Bangalore]

32. What will be the O/p of the below code Snippet?

```
import java.lang.*;
public class NewClass
{
        public static void main(String args[])
        {
                //Use of signum() method
                double x = 10.4556, y = 23.34789;
                double signm = Math.signum(x);
                System.out.println(“Signum of 10.4556 = “+signm);

                signm = Math.signum(y);
                System.out.println(“Signum of 23.34789 = “+signm);
                System.out.println(“”);

                //Use of round() method
                double r1 = Math.round(x);
                System.out.println(“Round off 10.4556 = “+r1);

                double r2 = Math.round(y);
                System.out.println(“Round off 23.34789 = “+r2);
                System.out.println(“”);
        }
}
```

1. Signum of 10.4556 = -1.0
   Signum of 23.34789 = 1.0
   Round off 10.4556 = 10.0
   Round off 23.34789 = 23.0
2. Signum of 10.4556 = 1.0
   Signum of 23.34789 = 0.0
   Round off 10.4556 = 10.0
   Round off 23.34789 = 23.0
3. Signum of 10.4556 = 1.0
   Signum of 23.34789 = 1.0
   Round off 10.4556 = 10.0

Round off 23.34789 = 23.0

4. Signum of 10.4556 = 1.0
   Signum of 23.34789 = -1.0
   Round off 10.4556 = 10.0
   Round off 23.34789 = 23.0

33. Which of the following statement/s are true about abstract classes in Java?
    1. Any concrete class (i.e. class without abstract keyword) that extends an abstract class must override all the abstract methods of the class
    2. These methods are sometimes referred to as subclass's responsibility because they have no implementation specified in the superclass
    3. Both 1 and 2
    4. None of the above

34. What will be the o/p of the following code snippet?

```
class Test
{
        public static void main(String args[])
        {
                for (int x = 0; x < 4; x++)
                {
                        System.out.print(x+ " ");
                }
                System.out.print(x);
        }
}
```

    1. 0 1 2 3 3
    2. 0 1 2 3 4
    3. 0 1 2 3
    4. Error

35. What will be the o/p of following code snippet?

```
import java.util.ArrayList;
import java.util.List;

public class CreateArrayListFromCollectionExample{
        public static void main(String[]args){
                List<Integer> firstFivePrimeNumbers = new ArrayList<>();
                firstFivePrimeNumbers.add(2);
                firstFivePrimeNumbers.add(3);
                firstFivePrimeNumbers.add(5);
                firstFivePrimeNumbers.add(7);
                firstFivePrimeNumbers.add(11);

                List<Integer> firstTenPrimeNumbers = new ArrayList<>(firstFivePrimeNumbers);

                List<Integer>nextFivePrimeNumbers = new ArrayList<>(firstFivePrimeNumbers);
                nextFivePrimeNumbers.add(9);
                nextFivePrimeNumbers.add(13);
                nextFivePrimeNumbers.add(17);
                nextFivePrimeNumbers.add(19);
                nextFivePrimeNumbers.add(23);

                firstTenPrimeNumbers.addAll(nextFivePrimeNumbers);
                System.out.println(firstTenPrimeNumbers);
        }
}
```

1. [2, 3, 5, 7, 11]
2. [2, 3, 5, 7, 11, 9, 13, 17, 19, 23]
3. [13, 17, 19, 23]
4. [3, 5, 7, 11, 13, 17, 19, 23]

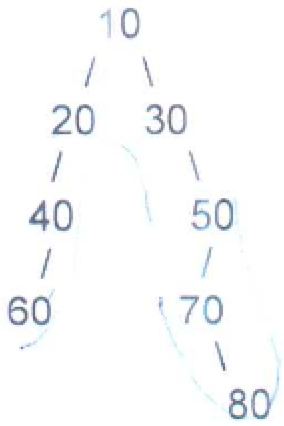
36. What will the Output of the following code snippet?

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

public class FailFastExample{
        public static void main(String[]args)
        {
                Map<String, String> cityCode = new HashMap<String, String>();
                cityCode.put(“Delhi”, “India”);
                cityCode.put(“Moscow”, “Russia”);
                cityCode.put(“New York”, “USA”);

                Iterator iterator = cityCode.keySet().iterator();
                while(iterator.hasNext(){
                        System.out.println(cityCode.get(iterator.next()));
                        cityCode.put(“Istanbul”, “Turkey”);
                }
        }
}
```

1. India
   Russia
   USA
   Turkey
2. India
   Russia
   USA
   Turkey
   Turkey
   Turkey
3. India
   Turkey
   Russia
   Turkey
   USA
   Turkey
4. India
   Exception in thread “main”

```
java.util.ConcurrentModificationException
at
java.util.HashMap$HashIterator.nextNode(HashMap.java:1442)
at
java.util.HashMap$KeyIterator.next(HashMap.java:1466)
at
FailFastExample.main(FailFastExample.java: 18)
```

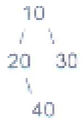
37. Given following tree, what will be the output of inorder traversal?



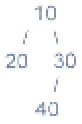
1. 40 60 20 10 30 70 80 50
2. 60 40 20 10 80 70 50 30
3. 60 40 20 10 70 30 80 50
4. 60 40 20 10 30 70 80 50

38. Which among the following is a complete binary tree?

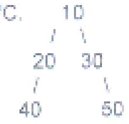
A.



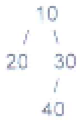
B.



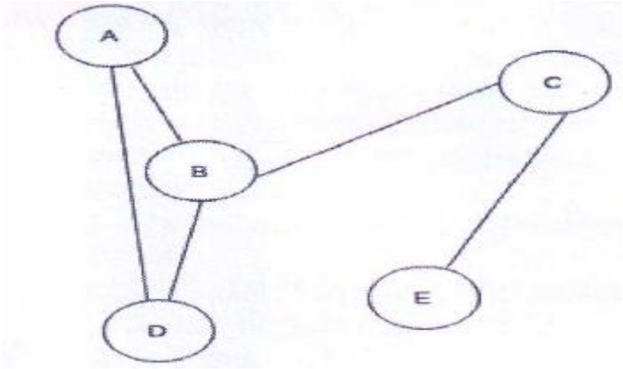
C.



D.



39. In the below given graph identify the cut vertices.



1. B and E
2. C and D
3. A and E
4. C and B

40. What is the time complexity of Binary Search?
    1. log2(n)
    2. Nlog2(n)
    3. O(n)
    4. O(1)