Deependra Singh Kushwah

Email: **deependrasingh191@gmail.com**	Mobile: **8904752330**
Qualification:	Specialization:
Current City: **#127, Deepak Apartments, 7th Cross, 12th A Main, BTM 1st Stage, Bangalore**	Preferred Location:
Secondary Skills:	Awards:
LinkedIn Profile:	

Rating (Max: 5): 0.71

Category Based Performance Summary				
Category Name	**Basic**	**Intermediate**	**Advanced**	**Percentage**
Basic Java	3/3	0/0	0/0	100.00
Collections	1/1	0/0	2/2	100.00
Threads	2/2	0/0	2/2	100.00
Data Structures and Algorithms	Marks Obtained: 0 / 60			0.00

Detailed Question Responses

Basic Java : Basic

What will be the output of the below program?

```
int i = 1, j = 10;
do
{
   if(i > j)
   {
      break;
   }
```

```
    j--;
} while (++i < 5);
System.out.println("i = " + i + " and j = " + j);
```

- i = 6 and j = 5
- i = 5 and j = 5
- i = 6 and j = 4
- **i = 5 and j = 6**

Collections : Advanced

What is the output for the below code ?

```
public class NameBean {
   private String str;

   NameBean(String str ){
      this.str = str;
   }

   public String toString() {
      return str;
   }
}

import java.util.HashSet;
public class CollClient {

   public static void main(String ... sss) {
      HashSet myMap = new HashSet();
      String s1 = new String("das");
      String s2 = new String("das");
   NameBean s3 = new NameBean("abcdef");
   NameBean s4 = new NameBean("abcdef");

      myMap.add(s1);
      myMap.add(s2);
      myMap.add(s3);
      myMap.add(s4);

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

- **das abcdef abcdef**

- das das abcdef abcdef
- das abcdef
- abcdef abcdef

Collections : Advanced

What will be the output of the program?

```
TreeSet map = new TreeSet();
map.add("one");
map.add("two");
map.add("three");
map.add("four");
map.add("one");
Iterator it = map.iterator();
while (it.hasNext() )
{
   System.out.print( it.next() + " " );
}
```

- one two three four
- four three two one
- **four one three two**
- one two three four one

Explanation: TreeSet is one of the Collection classes. It lets you access the elements in your collection by key, or sequentially by key. It has considerably more overhead than ArrayList or HashMap.TreeSet keeps the elements in order at all times. With ArrayList you just sort when you need to. With TreeSets the key must be embedded in the object you store in the collection.TreeSet and its brother TreeMap oddly have nothing to do with representing trees. Internally they use a tree organisation to give you an alphabetically sorted Set/Map, but you have no control over links between parents and children. Hence following the alphabatical order output will be, four one three two .

Collections : Basic

What will get printed?

```
final List intList=newArrayList();
intList.add(1);
intList.add(2);
intList.add(3);
intList.add(4);
intList.add(5);
intList.remove(2);
System.out.println(intList.get(2));
```

- 3
- 2
- **4**
- java.lang.IndexOutOfBoundsException
- java.lang.NullPointerException

Threads : Basic

In the below class, can 2 different threads access method1 and method2 simultaneously?

```
class TestThread{
   synchronized static void method1(){ ... }
   synchronized static void method2(){ ... }
}
```

- Yes
- **No**

Threads : Advanced

What is the output for the below code?

```
public class B {
public static synchronized void printName(){
try{
```

```
System.out.println("printName");
Thread.sleep(5*1000);
}catch(InterruptedException e){
}
}
public synchronized void printValue(){
System.out.println("printValue");
}
}
public class Test extends Thread{
B b = new B();
public static void main(String argv[]) throws Exception{
Test t = new Test();
Thread t1 = new Thread(t,"t1");
Thread t2 = new Thread(t,"t2");
t1.start();
t2.start();
}
public void run(){
if(Thread.currentThread().getName().equals("t1")){
b.printName();
}else{
b.printValue();
}
}
```

- print : printName , then wait for 5 seconds then print : printValue
- **print : printName then print : printValue**
- print : printName then wait for 5 minutes then print : printValue
- Compilation succeed but Runtime Exception

Basic Java : Basic

What is the result of below code?

```
public int method1() {
    try{
        return 2;
    }finally{
        return 5;
    }
}
```

- Compilation fails as there is no catch block

- Compilation fails as there are multiple return statements
- On invocation, the method returns 2
- **On invocation, the method returns 5**
- Throws a RuntimeException

Explanation:

Threads : Advanced

What will be the output of the below program?

```
class s1 implements Runnable
{
   int x = 0, y = 0;
   int addX() {x++; return x;}
   int addY() {y++; return y;}
   public void run() {
   for(int i = 0; i < 10; i++)
      System.out.println(addX() + " " + addY());
}
   public static void main(String args[])
   {
      s1 run1 = new s1();
      s1 run2 = new s1();
      Thread t1 = new Thread(run1);
      Thread t2 = new Thread(run2);
      t1.start();
      t2.start();
   }
}
```

- Compile time Error: There is no start() method
- Will print in this order: 1 1 2 2 3 3 4 4 5 5...
- **Will print but not exactly in an order (e.g: 1 1 2 2 1 1 3 3...)**
- Will print in this order: 1 2 3 4 5 6... 1 2 3 4 5 6...

Threads : Basic

What is the output for the below code ?

```
class A implements Runnable{
   public void run(){
      System.out.println(Thread.currentThread().getName());
   }
}
1. public class Test {
2.    public static void main(String... args) {
3.       A a = new A();
4.       Thread t = new Thread(a);
5.       t.setName("good");
6.       t.start();
7.    }
8. }
```

- ✔ **good**
- null
- Compilation fails with an error at line 5
- Compilation succeed but Runtime Exception

Basic Java : Basic

What will be the output of the below program?

```
public class TestDogs
{
   public static void main(String [] args)
   {
      Dog [][] theDogs = new Dog[3][];
      System.out.println(theDogs[2][0].toString());
   }
}
class Dog { }
```

- null
- theDogs
- Compilation fails
- ✔ **An exception is thrown at runtime**

Merge sorted Linked List

Problem Statement

Merge two sorted linked list. Two sorted singly linked list with head pointers head1 and head2 are given, merge them in sorted order and return head of the newly created linked list.

Constraints:

- You need to just implement the provided method i.e. **mergeLinkedList()**.
- In input of test cases, two lists will be separated by colon(:). Elements in each list will be separated by a space.
- You just need to return the reference in head node of merged linked list.

Sample Tests

#	Input	Output	Explanation
1	1 3 5 7 9:2 4 6 8 10	1 2 3 4 5 6 7 8 9 10	

Solution

```
#include
struct node{
    int value;
    struct node * next;
};
typedef struct node Node;
Node * mergeLinkedList(Node * head1, Node * head2){
    //Your code here. Return head of merged linked list.
            //Your code here.
        if(head1 == null)
            return head2;
        else if(head2 == null)
            return head1;
        Node *head=null;
        if(head1->value > head2->value){
            head = head2;
            head2 = head2->next;
        }
        else{
             head = head1;
             head1 = head1->next;
        }

        Node *returnHead = head;

        do{
            if((head1->value) > (head2->value)){
            head->next = head2;
            head2 =head2->next;
            }
        else{
            head->next = head1;
            head1=head1->next;
        }
        }
        while((head1!=null)&&(head2!=null));
        if(head1!=null)
            head->next = head1;
        else if(head2!=null)
            head->next=head2;

    }
    return returnHead;
}
```

Result
Error:prog.c: In function 'mergeLinkedList': prog.c:14:15: error: 'null' undeclared (first use in this function) if(head1 == null) ^ prog.c:14:15: note: each undeclared identifier is reported only once for each function it appears in prog.c:28:9: warning: unused variable 'returnHead' [-Wunused-variable] Node *returnHead = head; ^ prog.c: At top level: prog.c:47:2: error: expected identifier or '(' before 'return' return returnHead; ^ prog.c:49:1: error: expected identifier or '(' before '}' token } ^ prog.c: In function 'create_node': prog.c:52:28: warning: implicit declaration of function 'malloc' [-Wimplicit-function-declaration] Node * temp = (Node *)malloc(sizeof(Node)); ^ prog.c:52:28: warning: incompatible implicit declaration of built-in function 'malloc' prog.c:52:28: note: include ' ' or provide a declaration of 'malloc' prog.c: In function 'readline': prog.c:73:20: warning: incompatible implicit declarat

Data Structures and Algorithms : Basic

Mirror the tree

Problem Statement

Given a Binary Search Tree, mirror it in place. You are provided with Node structure in the code. You need to implement the provided method which takes the root node of a BST as parameter and returns the root node of mirrored tree.

Constraints:

- You need to just implement the provided method i.e. **mirrorTheTree()**.
- Input in test case will be the sequence in which the nodes are inserted to create a binary search tree.
- Output of a test case is represented by inorder sequence of mirrored tree.

Sample Tests

#	Input	Output	Explanation
1	8 9 11 1 5 3 4 2	11 9 8 5 4 3 2 1	

Solution

Language : C

```
struct node{
    int value;
    struct node * left;
    struct node * right;
};
typedef struct node Node;
Node * mirrorTheTree(Node * root){
    //Your code here
    if(root == null)
    return null;

    root->left = mirrorTheTree(root->right);
    root->right = mirrorTheTree(root->left);
    return root;
}
```

Result

Error:prog.c: In function 'mirrorTheTree': prog.c:13:13: error: 'null' undeclared (first use in this function) if(root == null) ^ prog.c:13:13: note: each undeclared identifier is reported only once for each function it appears in prog.c: In function 'create_node': prog.c:21:28: warning: implicit declaration of function 'malloc' [-Wimplicit-function-declaration] Node * temp = (Node *)malloc(sizeof(Node)); ^ prog.c:21:28: warning: incompatible implicit declaration of built-in function 'malloc' prog.c:21:28: note: include '' or provide a declaration of 'malloc' prog.c: In function 'readline': prog.c:43:20: warning: incompatible implicit declaration of built-in function 'malloc' char * line = malloc(100), * linep = line; ^ prog.c:43:20: note: include '' or provide a declaration of 'malloc' prog.c:55:28: warning: implicit declaration of function 'realloc' [-Wimplicit-function-declaration]

Reverse Linked List

Problem Statement

Write a program to reverse a given singly linked list. You will need to implement the provided method which will take head node of list as parameter. Return value should be the head of reversed linked list

Constraints:

- You need to just implement the provided method i.e. **reverseLinkedList()**.
- Input Test case represents the space separated elements of linked list starting from left to right.

Sample Tests			
#	**Input**	**Output**	**Explanation**
1	12 9 19 34 14 5	5 14 34 19 9 12	

Solution

```
struct node{
    int value;
    struct node * next;
};
typedef struct node Node;
Node * reverseLinkedList(Node * head){
    //Your code here.
    Node *returnHead = null;
    Node *prev, *curr, *next;
    prev =null;
    curr = head;
    next = curr->next;
    while(curr->next!=null)
    {
        curr->next = prev;
        prev =curr;
        curr =next;
        next = curr->next;

    }
}
```

Result

Error:prog.c: In function 'reverseLinkedList': prog.c:13:21: error: 'null' undeclared (first use in this function) Node *returnHead = null; ^ prog.c:13:21: note: each undeclared identifier is reported only once for each function it appears in prog.c:13:8: warning: unused variable 'returnHead' [-Wunused-variable] Node *returnHead = null; ^ prog.c: In function 'create_node': prog.c:29:28: warning: implicit declaration of function 'malloc' [-Wimplicit-function-declaration] Node * temp = (Node *)malloc(sizeof(Node)); ^ prog.c:29:28: warning: incompatible implicit declaration of built-in function 'malloc' prog.c:29:28: note: include '' or provide a declaration of 'malloc' prog.c: In function 'readline': prog.c:50:20: warning: incompatible implicit declaration of built-in function 'malloc' char * line = malloc(100), * linep = line; ^ prog.c:50:20: note: include '' or provide a declarati