Following numbers inserted into an empty Binary Search Tree:

10,1,3,5,15,12,17,18

What will be the height of this BST?

Note : Height is the max distance of a leaf node from the root

Map.put(key k , value v) return

Stream words = Stream.of("one", "two", "three");

int len = words.mapToInt(String::length).reduce(0, (len1, len2) -> len1 + len2);

System.out.println(len);

String s1 = "abc";

String s2 = s1;

s1 += "d";

System.out.println(s1 + " " + s2 + " " + (s1 == s2));

public class Test

{

public static void main(String[] args) throws InterruptedException

{

String str = new String("Morgan Stanley");

str = null;

System.gc();

Thread.sleep(1000);

System.out.println("end of main");

}

@Override

protected void finalize()

{

System.out.println("finalize method called");

}

}

public class RTExcept {

2. public static void throwit () {

3. System.out.print("throwit ");

4. throw new RuntimeException();

5. }

6. public static void main(String [] args) {

7. try {

8. System.out.print("hello ");

9. throwit();

10. }

11. catch (Exception re ) {

12. System.out.print("caught ");

13. }

14. finally {

15. System.out.print("finally ");

16. }

17. System.out.println("after ");

18. }

19. }

class PassA {

2. public static void main(String [] args) {

3. PassA p = new PassA();

4. p.start();

5. }

6.

7. void start() {

8. long [] a1 = {3,4,5};

9. long [] a2 = fix(a1);

10. System.out.print(a1[0] + a1[1] + a1[2] + " ");

11. System.out.println(a2[0] + a2[1] + a2[2]);

12. }

13.

14. long [] fix(long [] a3) {

15. a3[1] = 7;

16. return a3;

17. }

18.}