**A+ Computer Science  
Maps M/C TEST**

**Directions ::** On your answer sheet, mark the letter of the best answer to each question.

1. Which of the following Map methods will add a <key,value> pair to a Map?

|  |  |
| --- | --- |
| a. | add() |
| b. | get() |
| c. | set() |
| d. | put() |
| e. | append() |

2. Which of the following TreeMap instantiations would create a TreeMap that could store String-Integer pairs?

|  |  |
| --- | --- |
| a. | TreeMap<String,Integer> map;  map = new TreeMap<String,Integer>(); |
| b. | TreeMap<Integer,String> map;  map = new TreeMap<Integer,String>(); |
| c. | TreeMap<Double,String> map;  map = new TreeMap<Double,String>(); |
| d. | TreeMap<Boolean,String> map;  map = new TreeMap<Boolean,String>(); |
| e. | TreeMap<Boolean,Byte> map;  map = new TreeMap<Boolean,Byte>(); |

3. Which of the following TreeMap instantiations would create an TreeMap that could store Double-String pairs?

|  |  |
| --- | --- |
| a. | TreeMap<String,Integer> map;  map = new TreeMap<String,Integer>(); |
| b. | TreeMap<Integer,String> map;  map = new TreeMap<Integer,String>(); |
| c. | TreeMap<Double,String> map;  map = new TreeMap<Double,String>(); |
| d. | TreeMap<Boolean,String> map;  map = new TreeMap<Boolean,String>(); |
| e. | TreeMap<Boolean,Byte> map;  map = new TreeMap<Boolean,Byte>(); |

4. Which of the following is an interface?

|  |  |
| --- | --- |
| a. | ArrayList |
| b. | HashSet |
| c. | TreeMap |
| d. | Collection |
| e. | LinkedList |

5. Which of the following is a child of SortedMap?

|  |  |
| --- | --- |
| a. | ArrayList |
| b. | HashMap |
| c. | TreeMap |
| d. | Map |
| e. | LinkedList |

6. What is the output of line 1?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("8",6);

map.put("8",3);

map.put("7",2);

map.put("5",9);

map.put("6",8);

out.println(map.put("8",5)); //line 1

|  |  |
| --- | --- |
| a. | 6 |
| b. | 3 |
| c. | 5 |
| d. | 2 |
| e. | null |

7. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("8",6);

map.put("8",3);

map.put("7",2);

map.put("5",9);

map.put("6",8);

map.put("9",2);

map.put("5",6);

map.put("2",8);

map.put("2",2);

out.println(map.get("5"));

|  |  |
| --- | --- |
| a. | 6 |
| b. | 3 |
| c. | 5 |
| d. | 2 |
| e. | null |

8. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("8",6);

map.put("8",3);

map.put("7",2);

map.put("5",9);

map.put("6",8);

map.put("9",2);

map.put("5",6);

map.put("2",8);

map.put("2",2);

out.println(map.get("2"));

|  |  |
| --- | --- |
| a. | 6 |
| b. | 3 |
| c. | 5 |
| d. | 2 |
| e. | null |

9. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("8",6);

map.put("8",3);

map.put("7",2);

map.put("5",9);

map.put("6",8);

map.put("9",2);

map.put("5",6);

map.put("2",8);

map.put("2",2);

out.println(map.get("1"));

|  |  |
| --- | --- |
| a. | 6 |
| b. | 3 |
| c. | 5 |
| d. | 2 |
| e. | null |

10. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

String[] list = "4 5 8 2 2 2 6".split(" ");

for(String s : list)

{

if(map.get(s)==null)

{

map.put(s,1);

}

else

{

map.put(s, map.get(s)+1);

}

}

out.println( map );

|  |  |
| --- | --- |
| a. | {2=3, 4=1, 5=1, 6=1, 8=1} |
| b. | {2=1, 4=1, 5=1, 6=1, 8=1} |
| c. | {2=1, 4=1, 5=1, 6=1, 8=1} |
| d. | {2=3, 4=2, 5=1, 6=1, 8=1} |
| e. | {2=3, 4=3, 5=1, 6=1, 8=1} |

11. What is the output by the code below?

Map<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {4,5,8,2,2,2,3,3,6,6};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

out.println(map);

|  |  |
| --- | --- |
| a. | {2=1, 3=1, 4=1, 5=1, 6=1, 8=1} |
| b. | {2=2, 3=2, 4=2, 5=1, 6=1, 8=1} |
| c. | {2=4, 3=1, 4=1, 5=1, 6=2, 8=1} |
| d. | {2=2, 3=2, 4=2, 5=1, 6=2, 8=1} |
| e. | {2=3, 3=2, 4=1, 5=1, 6=2, 8=1} |

12. What is the output?

Map<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {2,3,4,2,3,4,5,3,2,3,4,2,3,4,2,3};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

out.println(map);

|  |  |
| --- | --- |
| a. | {2=4, 3=7, 4=4, 5=1} |
| b. | {2=5, 3=6, 4=4, 5=1} |
| c. | {2=3, 3=7, 4=5, 5=1} |
| d. | {2=1, 3=1, 4=1, 5=1} |
| e. | {2=2, 3=8, 4=4, 5=2} |

13. Which of the following would correctly fill < \*1 > ?

Scanner kb = new Scanner(in);

Map<Integer,String> map;

map = new < \*1 > map.put(1,"one");

map.put(2,"two");

map.put(3,"three");

map.put(4,"four");

map.put(5,"five");

map.put(6,"six");

map.put(7,"seven");

map.put(8,"eight");

map.put(9,"nine");

map.put(10,"ten");

for(int i=0; i<10; i++)

{

out.print("Enter a integer [1-10] :: ");

int num = < \*2 >

out.println(< \*3 > );

}

|  |  |
| --- | --- |
| a. | TreeMap<String>(); |
| b. | TreeMap<Integer,String>(); |
| c. | TreeMap<String, Integer>(); |
| d. | TreeMap<Integer>(); |
| e. | TreeMap<Integer, Integer>(); |

14. Assuming that < \*1 > was filled correctly, which of the following would correctly

fill < \*2 > ?

Scanner kb = new Scanner(in);

Map<Integer,String> map;

map = new < \*1 > map.put(1,"one");

map.put(2,"two");

map.put(3,"three");

map.put(4,"four");

map.put(5,"five");

map.put(6,"six");

map.put(7,"seven");

map.put(8,"eight");

map.put(9,"nine");

map.put(10,"ten");

for(int i=0; i<10; i++)

{

out.print("Enter a integer [1-10] :: ");

int num = < \*2 >

out.println(< \*3 > );

}

|  |  |
| --- | --- |
| a. | kb.nextDouble(); |
| b. | kb.nextFloat(); |
| c. | kb.nextLong(); |
| d. | kb.nextInt(); |
| e. | kb.nextChar(); |

15. Assuming that < \*1 > and < \*2 > are filled correctly, which of the following would correctly fill < \*3 > ?

Scanner kb = new Scanner(in);

Map<Integer,String> map;

map = new < \*1 > map.put(1,"one");

map.put(2,"two");

map.put(3,"three");

map.put(4,"four");

map.put(5,"five");

map.put(6,"six");

map.put(7,"seven");

map.put(8,"eight");

map.put(9,"nine");

map.put(10,"ten");

for(int i=0; i<10; i++){

out.print("Enter a integer [1-10] :: ");

int num = < \*2 >

out.println(< \*3 > );

}

|  |  |
| --- | --- |
| a. | map.get(num) |
| b. | map.put(num) |
| c. | map.get(i) |
| d. | map.put(i) |
| e. | map.set(i) |

16. Which of the following best matches the runtime for TreeMap?

put() get()

LINE 1 O(1) O(N)

LINE 2 O(1) O(1)

LINE 3 O(N) O(log2N)

LINE 4 O(log2N) O(log2N)

LINE 5 O(log2N) O(N)

|  |  |
| --- | --- |
| a. | Line 1 |
| b. | Line 2 |
| c. | Line 3 |
| d. | Line 4 |
| e. | Line 5 |

17. Which of the following would correctly fill **<\*1>** ?

public class Histogram

{

private Map<String,Integer> map;

public Histogram( String list )

{

map = new TreeMap<String,Integer>();

String[] wrds = list.split(" ");

for( String word : wrds )

{

if( **<\*1>** )

{

map.put(word,1);

}

else

{

int cnt = map.get(word);

**<\*2>**

}

}

}

public void printStats()

{

for( **<\*3>** )

{

out.println( s + " has a count of " + **<\*4>**

}

}

}

|  |  |
| --- | --- |
| a. | map.get(word)==null |
| b. | map.containsKey(word)==false |
| c. | map.get(word)!=null |
| d. | map.containsKey(word)==true |
| e. | A and B only |

18. Assuming **<\*1>** has been filled correctly, which of the following would correctly fill **<\*2>** ?

public class Histogram

{

private Map<String,Integer> map;

public Histogram( String list )

{

map = new TreeMap<String,Integer>();

String[] wrds = list.split(" ");

for( String word : wrds )

{

if( **<\*1>** )

{

map.put(word,1);

}

else

{

int cnt = map.get(word);

**<\*2>**

}

}

}

public void printStats()

{

for( **<\*3>** )

{

out.println( s + " has a count of " + **<\*4>**

}

}

}

|  |  |
| --- | --- |
| a. | map.put(word,cnt); |
| b. | map.put(word,cnt-1); |
| c. | map.put(word,cnt++); |
| d. | map.put(word,cnt+1); |
| e. | None of these |

19. Assuming **<\*1>** and **<\*2>** have been filled correctly, which of the following would correctly fill **<\*3>** ?

public class Histogram

{

private Map<String,Integer> map;

public Histogram( String list )

{

map = new TreeMap<String,Integer>();

String[] wrds = list.split(" ");

for( String word : wrds )

{

if( **<\*1>** )

{

map.put(word,1);

}

else

{

int cnt = map.get(word);

**<\*2>**

}

}

}

public void printStats()

{

for( **<\*3>** )

{

out.println( s + " has a count of " + **<\*4>**

}

}

}

|  |  |
| --- | --- |
| a. | String s : keySet().iterator() |
| b. | String s : map.keySet() |
| c. | String s : map.iterator() |
| d. | String s : map.key().iterator() |
| e. | String s : keySet().map.iterator() |

20. Assuming **<\*1>, <\*2>,** and **<\*3>** have been filled correctly, which of the following would correctly fill **<\*4>** ?

public class Histogram

{

private Map<String,Integer> map;

public Histogram( String list )

{

map = new TreeMap<String,Integer>();

String[] wrds = list.split(" ");

for( String word : wrds )

{

if( **<\*1>** )

{

map.put(word,1);

}

else

{

int cnt = map.get(word);

**<\*2>**

}

}

}

public void printStats()

{

for( **<\*3>** )

{

out.println( s + " has a count of " + **<\*4>**

}

}

}

|  |  |
| --- | --- |
| a. | map.get(it)); |
| b. | map.get(0)); |
| c. | map.get(s)); |
| d. | map.get(it.next())); |
| e. | map.set(s)); |

21. Which of the following would correctly fill **< \*1 >** ?

public class Lines

{

private Map<String,Set<String>> map;

public Lines()

{

map = new TreeMap<String,Set<String>>();

}

public void setPerson(String pair)

{

String[] list = pair.split(" ");

String line = list[0];

String person = list[1];

Set<String> s;

if( **< \*1 >**  ){

s = map.get(line);

s.add(person);

**< \*3 >**

}

else{

s = **< \*2 >**

s.add(person);

**< \*3 >**

}

}

public String toString()

{

String output="";

for( **< \*4>**  )

{

output+=key+" "+map.get(key);

output+="\n";

}

return output;

}

}

|  |  |
| --- | --- |
| a. | map.get(line)!=null |
| b. | map.containsKey(person)==false |
| c. | map.get(person)!=null |
| d. | map.containsKey(line)=false |
| e. | more than one of these |

22. Assuming that **< \*1 >** has been filled correctly, which of the following would correctly fill **< \*2 >** ?

public class Lines

{

private Map<String,Set<String>> map;

public Lines(){

map = new TreeMap<String,Set<String>>();

}

public void setPerson(String pair){

String[] list = pair.split(" ");

String line = list[0];

String person = list[1];

Set<String> s;

if( **< \*1 >**  ){

s = map.get(line);

s.add(person);

**< \*3 >**

}

else{

s = **< \*2 >**

s.add(person);

**< \*3 >**

}

}

public String toString(){

String output="";

for( **< \*4>**  )

{

output+=key+" "+map.get(key);

output+="\n";

}

return output;

}

}

|  |  |
| --- | --- |
| a. | new Set<String>(); |
| b. | new TreeSet<>(); |
| c. | new TreeSet<String>(); |
| d. | new Set(); |
| e. | None of these |

23. Assuming that **< \*1 >** and **< \*2 >**  have been filled correctly, which of the following would correctly fill **< \*3 >** ?

public class Lines

{

private Map<String,Set<String>> map;

public Lines(){

map = new TreeMap<String,Set<String>>();

}

public void setPerson(String pair){

String[] list = pair.split(" ");

String line = list[0];

String person = list[1];

Set<String> s;

if( **< \*1 >**  ){

s = map.get(line);

s.add(person);

**< \*3 >**

}

else{

s = **< \*2 >**

s.add(person);

**< \*3 >**

}

}

public String toString()

{

String output="";

for( **< \*4>**  )

{

output+=key+" "+map.get(key);

output+="\n";

}

return output;

}

}

|  |  |
| --- | --- |
| a. | map.put(line,person); |
| b. | map.put(line,map); |
| c. | map.put(line,line); |
| d. | map.put(line,s); |
| e. | None of these |

24. Assuming that **< \*1 >, < \*2 >** and **< \*3>**  have been filled correctly, which of the following would correctly fill **< \*4 >** ?

public class Lines

{

private Map<String,Set<String>> map;

public Lines(){

map = new TreeMap<String,Set<String>>();

}

public void setPerson(String pair){

String[] list = pair.split(" ");

String line = list[0];

String person = list[1];

Set<String> s;

if( **< \*1 >**  ){

s = map.get(line);

s.add(person);

**< \*3 >**

}

else{

s = **< \*2 >**

s.add(person);

**< \*3 >**

}

}

public String toString(){

String output="";

for( **< \*4>**  )

{

output+=key+" "+map.get(key);

output+="\n";

}

return output;

}

}

|  |  |
| --- | --- |
| a. | String key : map |
| b. | String key : map.keySet().iterator() |
| c. | String key : map.key() |
| d. | String key : map.set() |
| e. | String key : map.keySet() |

25. Which of the following interfaces must be implemented by all Objects stored in a TreeMap?

|  |  |
| --- | --- |
| a. | Locatable |
| b. | Sortable |
| c. | Treeable |
| d. | Mapable |
| e. | Comparable |

26. Which of the following best matches the runtime for HashMap assuming the hash function is constant runtime with few collisions?

put() get()

LINE 1 O(N2) O(N2)

LINE 2 O(1) O(1)

LINE 3 O(N) O(log2N)

LINE 4 O(log2N) O(log2N)

LINE 5 O(log2N) O(N)

|  |  |
| --- | --- |
| a. | Line 1 |
| b. | Line 2 |
| c. | Line 3 |
| d. | Line 4 |
| e. | Line 5 |

27. Which is a Map?

|  |  |
| --- | --- |
| a. | A collection with no duplicates |
| b. | A resizeable array |
| c. | A list where each element points to the next |
| d. | A collection of key value pairs |
| e. | A LIFO list |

28. Which one of these functions returns a Set of map’s keys?

|  |  |
| --- | --- |
| a. | keySet() |
| b. | entrySet() |
| c. | values() |
| d. | keys() |
| e. | toArray() |

29. Which one of these functions returns a Set of all the mapping entries?

|  |  |
| --- | --- |
| a. | keySet() |
| b. | entrySet() |
| c. | values() |
| d. | keys() |
| e. | toArray() |

30. Which one of these functions returns a Collection of all the values contained in a map?

|  |  |
| --- | --- |
| a. | keySet() |
| b. | entrySet() |
| c. | values() |
| d. | keys() |
| e. | toArray() |

31. What is the output by the code below?

Map<String,Integer> a, b;

a = new TreeMap<String,Integer>();

b = new TreeMap<String,Integer>();

a.put("8", 6);

b.put("8", 6);

out.println(a.equals(b));

|  |  |
| --- | --- |
| a. | 0 |
| b. | 1 |
| c. | true |
| d. | false |
| e. | null |

32. What is the output by the code below?

Map<String,Integer> a, b;

a = new TreeMap<String,Integer>();

b = new TreeMap<String,Integer>();

a.put("8", 3);

b.put("8", 4);

out.println(a.equals(b));

|  |  |
| --- | --- |
| a. | 0 |
| b. | 1 |
| c. | true |
| d. | false |
| e. | null |

33. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("foo", 12);

map.put("foo", 10);

out.println(map.containsValue(12));

|  |  |
| --- | --- |
| a. | 0 |
| b. | 1 |
| c. | true |
| d. | false |
| e. | null |

34. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("foo", 12);

map.put("foo", 10);

out.println(map.containsValue(10));

|  |  |
| --- | --- |
| a. | 0 |
| b. | 1 |
| c. | true |
| d. | false |
| e. | null |

35. What is the output by the code below?

Map<String,Integer> map;

map = new TreeMap<String,Integer>();

map.put("foo", 12);

map.put("foo", 10);

out.println(map.containsKey("foo"));

|  |  |
| --- | --- |
| a. | 0 |
| b. | 1 |
| c. | true |
| d. | false |
| e. | null |

36. What is the output?

TreeMap<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {9,2,3,4,5,6,7,10,3,4,2,3,4,2,3};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

System.out.println(map.values());

|  |  |
| --- | --- |
| a. | [3, 4, 3, 1] |
| b. | [3, 4, 3, 1, 1, 1, 1, 1] |
| c. | [4, 3, 1, 1, 1, 1, 1] |
| d. | [2, 3, 4, 5, 6, 7, 9] |
| e. | [2, 3, 4, 5, 6, 7, 9, 10] |

37. What is the output?

TreeMap<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {9,2,3,4,5,6,7,10,3,4,2,3,4,2,3};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

System.out.println( map.keySet() );

|  |  |
| --- | --- |
| a. | [3, 4, 3, 1] |
| b. | [3, 4, 3, 1, 1, 1, 1, 1] |
| c. | [4, 3, 1, 1, 1, 1, 1] |
| d. | [2, 3, 4, 5, 6, 7, 9] |
| e. | [2, 3, 4, 5, 6, 7, 9, 10] |

38. What is the output?

TreeMap<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {9,2,3,4,5,6,7,10,3,4,2,3,4,2,3};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

System.out.println( map.firstKey() );

|  |  |
| --- | --- |
| a. | 2 |
| b. | 3 |
| c. | 7 |
| d. | 9 |
| e. | 10 |

39. What is the output?

TreeMap<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {9,2,3,4,5,6,7,10,3,4,2,3,4,2,3};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

System.out.println( map.lastKey() );

|  |  |
| --- | --- |
| a. | 2 |
| b. | 3 |
| c. | 7 |
| d. | 9 |
| e. | 10 |

40. What is the output?

TreeMap<Integer,Integer> map;

map = new TreeMap<Integer,Integer>();

Integer[] list = {9,2,3,4,5,6,7,10,3,4,2,3,4,2,3};

for(Integer num : list)

if(map.get(num)==null)

map.put(num,1);

else

map.put(num,map.get(num)+1);

System.out.println( map.lastEntry() );

|  |  |
| --- | --- |
| a. | 2=3 |
| b. | 3=4 |
| c. | 7=1 |
| d. | 9=1 |
| e. | 10=1 |