Question 1

Not yet answered

Marked out of 1.00

Which of the following traversal methods can be used to print all nodes of a binary tree level by level?

Select one:

- a. **Post-order traversal**
- b. **Breadth-First Search** (selected)
- c. **Depth-First Search**
- d. **In-order traversal**

Question 2

Not yet answered

Marked out of 1.00

Which of the following is not a linear data structure?

Select one:

- a. **Stack**
- b. **Graph** (selected)
- c. **Array**
- d. **Queue**

Question 3

Not yet answered

Marked out of 1.00

Which data structure is best suited for implementing an LRU (Least Recently Used) cache?

Select one:

- a. **Stack**
- b. **Priority Queue**
- c. **Queue**
- d. **HashMap + Doubly Linked List** (selected)

Question 4

Not yet answered

Marked out of 1.00

Given:

```
3. public class Dark {
4.   int x = 3;
5.   public static void main(String[] args) {
6.     new Dark().go1();
7.   }
8.   void go1() {
9.     int x;
10.     go2(++x);
11.   }
12.   void go2(int y) {
13.     int x = ++y;
14.     System.out.println(x);
15.   }
16. }
```

What is the result?

Select one:

- a. 3
- b. 4
- c. 2
- d. 5
- e. An exception is thrown at runtime
- f. Compilation fails

Question 5

Not yet answered

Marked out of 1.00

What is the worst-case time complexity of inserting an element into a heap?

Select one:

- a. O(n)
- b. O(n log n)
- c. O(log n)
- d. O(1)

Question 6

Not yet answered

Marked out of 1.00

Which of the following is the most efficient way to check if any orders exist for a specific customer?

```
SELECT 1
FROM orders
WHERE customer_id = 101
LIMIT 1;
```

Select one:

- **a. Efficient, as it stops after finding the first match**
- **b. Inefficient, since it does a full table scan**
- **c. In syntax**
- **d. Inefficient, as it returns all matches**

Question 7

Not yet answered

Marked out of 1.00

Given:

```
3. public class Clumsy {
4.   public static void main(String[] args) {
5.     int j = 7;
6.     assert(++j > 7);
7.     assert(++j > 8): "hi";
8.     assert(j > 10): j=12;
9.     assert(j==12): doStuff();
10.      assert(j==12): new Clumsy();
11.  }
12.  static void doStuff() { }
13. }
```

Which are true? (Choose all that apply.)

Select one:

- **a. Compilation succeeds**
- **b. Compilation fails due to an error on line 6**
- **c. Compilation fails due to an error on line 7**
- **d. Compilation fails due to an error on line 10**
- **e. Compilation fails due to an error on line 8**
- **f. Compilation fails due to an error on line 9**

Question 8

Not yet answered

Marked out of 1.00

What is the time complexity to search for an element in a balanced Binary Search Tree (BST)?

Select one:

- a. **O(n log n)**
- b. **O(n)**
- c. **O(log n)**
- d. **O(1)**

Question 9

Not yet answered

Marked out of 1.00

What does the following query do?

```
SELECT e.name, d.name AS department
FROM employees e
LEFT JOIN departments d ON e.dept_id = d.id;
```

Select one:

- a. **Returns employees and their departments, including those without a department**
- b. **Returns only employees who belong to a department**
- c. **Returns only employees who are in multiple departments**
- d. **Returns departments and their employees**

Question **10**

Not yet answered

Marked out of 1.00

Given:

```
3. interface Vessel { }
4. interface Toy { }
5. class Boat implements Vessel { }
6. class Speedboat extends Boat implements Toy { }
7. public class Tree {
8.   public static void main(String[] args) {
9.     String s = "0";
10.     Boat b = new Boat();
11.    Boat b2 = new Speedboat();
12.     Speedboat s2 = new Speedboat();
13.     if((b instanceof Vessel) && (b2 instanceof Toy))  s += "1";
14.     if((s2 instanceof Vessel) && (s2 instanceof Toy)) s += "2";
15.     System.out.println(s);
16.   }
17. }
```

What is the result?

Select one:

- **a. An exception is thrown at runtime**
- **b. Compilation fails**
- **c. 02**
- **d. 012**
- **e. 01**
- **f. 0**

Question **11**

Not yet answered

Marked out of 1.00

What is the result of the following subquery usage?

```
SELECT name
FROM products
WHERE category_id = (
SELECT category_id
FROM categories
WHERE name = 'Electronics'
);
```

Select one:

- **a. Products in all categories**
- **b. Categories without products**
- **c. Products in the 'Electronics' category**
- **d. Products not in Electronics**

Question **12**

Not yet answered

Marked out of 1.00

What will be the output of the following SQL query?

```
SELECT department_id, COUNT(*) AS total_employees
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 5;
```

Select one:

- **a. All employees in all departments**
- **b. Employees with salary > 5**
- **c. Departments with more than 5 employees**
- **d. Departments where at least one employee exists**

Question **13**

Not yet answered

Marked out of 1.00

Which query returns the second highest salary from the employees table?

```
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

Select one:

- **a. Returns the second highest salary**
- **b. Returns the average of top 2 salaries**
- **c. Returns the highest salary**
- **d. Will throw an error due to nested query**

Question **14**

Not yet answered

Marked out of 1.00

Given:

```
3. class A { }
4. class B extends A { }
5. public class ComingThru {
6.   static String s = "-";
7.   public static void main(String[] args) {
8.     A[] aa = new A[2];
9.     B[] ba = new B[2];
10.      sifter(aa);
11.     sifter(ba);
12.      sifter(7);
13.      System.out.println(s);
14.  }
15.  static void sifter(A[]... a2)    { s += "1"; }
16.  static void sifter(B[]... b1)    { s += "2"; }
17.  static void sifter(B[] b1)       { s += "3"; }
18.  static void sifter(Object o)     { s += "4"; }
19. }
```

What is the result?

Select one:

- **a. Compilation fails**
- **b. -434**
- **c. -444**
- **d. -424**
- **e. -134**
- **f. -124**

Question **15**

Not yet answered

Marked out of 1.00

Given:

```
3. class Beta { }
4. class Alpha {
5.   static Beta b1;
6.   Beta b2;
7. }
8. public class Tester {
9.   public static void main(String[] args) {
10.    Beta b1 = new Beta();     Beta b2 = new Beta();
11.   Alpha a1 = new Alpha();   Alpha a2 = new Alpha();
12.    a1.b1 = b1;
13.    a1.b2 = b1;
14.    a2.b2 = b2;
15.    a1 = null;  b1 = null;  b2 = null;
16.    // do stuff
17.  }
18. }
```

When line 16 is reached, how many objects will be eligible for garbage collection?

Select one:

- a. 3
- b. 5
- c. 4
- d. 0
- e. 1
- f. 2

Quiz Navigation

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Finish attempt ... (http://kmitonline.com/mod/quiz/summary.php?attempt=480721)