CPT204 2024
Resit Exam Question

Short Answer Question (60 marks, 2 for each)

1. In the following code, ‘number’ is a _____ variable as it stores the actual values of primitive data types directly, while ‘message’ is a _____ variable as it stores the address of an object in memory.

```
public class Example {
    public static void main(String[] args) {
        int number = 42;
        String message = "Hello, world!";
    }
}
```

2. A _____ is a class from which other classes are derived, while a _____ is a class that is derived from another class.

3. In Java, _____ equality checks if two variables refer to the exact same object in memory, while _____ equality checks if two objects have the same state or content, as defined by the equals() method.

4. An _____ can have instance methods that implement a default behavior, while an _____ cannot have instance methods.

5. For scenarios requiring efficient random access through an index without frequent insertion or removal of elements except at the end of the list, the _____ class is more suitable. In contrast, for applications that frequently insert or delete elements at the beginning of the list, the _____ class is a better choice.

6. In the program below, a compilation error occurs at line 4 because the Fruit class does not implement the _____ interface, which is required for the Arrays.sort() method to sort the array of Fruit objects.

```
1.public class Test {
2.      public static void main(String[] args) {
3.          Fruit[] fruits = {new Fruit(2), new Fruit(3), new Fruit(1)};
4.          Arrays.sort(fruits);
5.      }
6.}

class Fruit {
   private double weight;
```

```
    public Fruit(double weight) {
        this.weight = weight;
    }
}
```

7. The following statement would encounter a compilation error because a generic type must be a _____ type. One way to correct the statement is to change the type parameter “int” to _____.

```
ArrayList<int> intList = new ArrayList<>();
```

8. The method header is left blank in the following code. Fill in the header using a generic method _____.

```
public class GenericMethodDemo {
    public static void main(String[] args ) {
    Integer[] numbers = {10, 20, 30, 40, 50};
    String[] cities = {"Tokyo", "London", "Paris", "New York"};
        print(numbers);
        print(cities);
    }

  public static_____ {
        for (int i = 0; i < list.length; i++)
            System.out.print(list[i] + " ");
        System.out.println();
    }
}
```

9. In the code below, ‘LinkedList’ is a _____ without specifying any type parameter, which implies it can hold elements of any type.

```
public class Example {
        public static void main(String[] args) {
                LinkedList list = new LinkedList();
                list.add("World");
                list.add(24);
                list.add(2.71);
        }
}
```

10. A list collection organizes elements in a _____ order, and permits specifying the position where an element is stored. Additionally, elements can be accessed by_____.

11. The _____ interface defines the natural ordering of objects of each class that implements it, whereas the _____ interface defines a separate class to compare objects of another class.

12. The _____ method in the Queue interface retrieves and removes the head of this queue, or null if this queue is empty. The _____ method in the Queue interface retrieves and removes the head of this queue and throws an exception if this queue is empty.

13. Suppose list list1 is [1, 3, 5] and list list2 is [2, 4, 6]. After list1.addAll(list2), list1 is _____.

14. In the following code, the key "101" will correspond to the value "_____" after all the operations.

```
public class Test {
   public static void main(String[] args) {
      Map<String, String> map = new HashMap<>();
      map.put("101", "Alice");
      map.put("102", "Bob");
      map.put("101", "Charlie");
   }
}
```

15. The output of the following code is _____.

```
HashMap<String, Integer> map = new HashMap<>();
map.put("One", 1);
map.put("Two", 2);
map.put("Three", 3);
System.out.println(map.get("Two"));
```

16. In the selection sort algorithm, the _____ element in the array is identified and exchanged with the _____ element in each iteration.

17. An algorithm with the O(n) time complexity is called a_____ algorithm and an algorithm with the O(log n) time complexity is called a _____ algorithm.

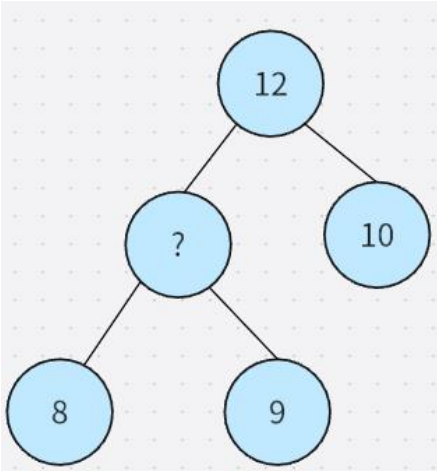
18. _____ is a method for solving complex problems by breaking them down into simpler subproblems where these subproblems overlap, unlike in _____ method where subproblems are more independent.

19. Use the Big O notation to estimate the time complexity of the following method: _____.

```
public static void mD(int[] m) {
   for (int i = 0; i < m.length; i++) {
      for (int j = 0; j < m.length; j++) {
         System.out.print(m[i] + " ");
      }
   }
}
```

20. Suppose a list is [12, 9, 15, 4]. After the second pass of bubble sort, the list becomes _____.

21. What value should best replace the '?' to maintain the max heap strictly?



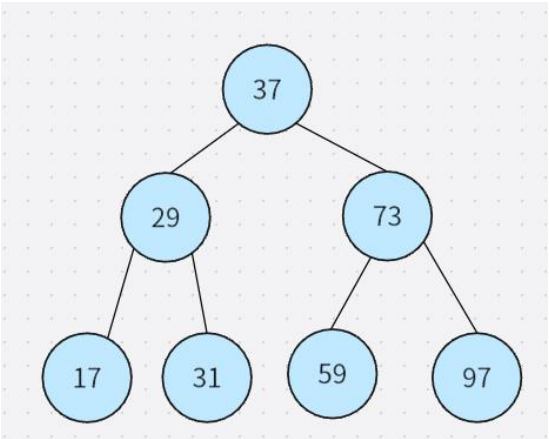
22. A _____ graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge, while in a _____ graph, not every pair of vertices is connected by an edge.

23. _____ is a graph traversal method that visits all the vertices of a graph as deeply as possible before backtracking, while _____ visits all the vertices of a graph level by level.

24. _____ is an algorithm in graph theory where the goal is to connect all vertices with the least total weight, while _____ is an algorithm where the goal is to find the path with the least total weight between two vertices.

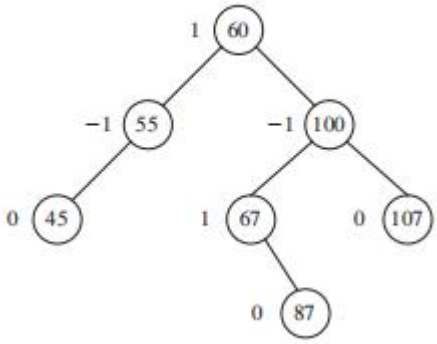
25. In a binary tree, the _____ traversal visits the root node before its children, while the _____ traversal visits the left subtree, then the root, and finally the right subtree.

26. The postorder traversal and the breadth-first traversal of the following tree are _____ and _____ respectively.



27. In an AVL tree, a _____ imbalance occurs when a node is inserted into the right subtree of the right child of A, while a _____ imbalance occurs when a node is inserted into the left subtree of the left child of A.

28. Given the original AVL tree below, after adding element 81, the _____rotation would be performed to rebalance the tree.



29. Collisions occur when two different keys produce the same _____. Besides open addressing, _____, where each hash table slot maintains a linked list of elements mapping to the same slot, is the other way to solve collisions.

30. _____ hashing uses a secondary hash function h' on the keys to determine the increments, aiming to avoid the _____ problem.

Coding Questions (40 marks, 20 for each)

Q1. You are developing a library management system that categorizes books into three sections: "Fiction," "Historical," and "Reference." Each book is placed in a queue according to its section for shelving.

Given three priority queues representing the books waiting to be shelved:

- Fiction Queue: {"The Great Gatsby", "To Kill a Mockingbird", "1984", "Pride

and Prejudice", "Harry Potter", "The Hobbit"}

- Historical Queue: {"Sapiens", "Pride and Prejudice", "A Brief History of Time", "Educated"}
- Reference Queue: {"The Great Gatsby", "A Brief History of Time", "Harry Potter", "Encyclopedia Britannica", "The Hobbit"}

You job is to find books that are present in all three sections (i.e. popular books). Complete the 3 tasks indicated in the following whiteboard.

Whiteboard (where students code answers):

```
import java.util.Arrays;
import java.util.HashSet;
import java.util.PriorityQueue;
import java.util.Set;

public class LibraryManagementSystem {

	public static void main(String[] args) {
		// Task 1: Create PriorityQueues for each section of books

		// Task 3: Call findPopularBooks() method and print out the name of the popular books
	}

	public static Set<String> findPopularBooks(
			PriorityQueue<String> fictionQueue,
			PriorityQueue<String> historicalQueue,
			PriorityQueue<String> referenceQueue) {

		// Task 2: Implement this method to find books that are present in all three sections

		return popularBooks;
	}
}
```

Q2. A human resource system needs to sort employee records based on different attributes. One common requirement is to sort employees by their ID numbers and another is to sort them by their names in a case-insensitive manner.

Given the following array of employee IDs and names:
Integer[] employeeIds = {102, 101, 104, 103, 105};
String[] employeeNames = {"John", "Alice", "bob", "Diana", "Charlie"};

You are required to use insertion sort algorithm to do the following 2 tasks as indicated in the white board.
White board (where students code answers):

```
import java.util.Comparator;

public class EmployeeRecordSorting {

    // Task 1: Use the following insertionSort() method with the Comparable
       interface to sort the array of employee IDs in ascending order.

       public static <E extends Comparable<E>> void insertionSort(E[] list) {

    }

    // Task 2: Use the following insertionSort() method with the Comparator
       interface to sort the array of employee names in a case-insensitive
       manner.

       public static <E> void insertionSort(E[] list, Comparator<? super E>
       comparator) {

       }

  public static void main(String[] args) {

          Integer[] employeeIds = {102, 101, 104, 103, 105};
          String[] employeeNames = {"John", "Alice", "bob", "Diana", "Charlie"};

          insertionSort(employeeIds);
          System.out.println("Sorted Employee IDs:");
          System.out.println(Arrays.toString(employeeIds));

          insertionSort(employeeNames,
String.CASE_INSENSITIVE_ORDER);
          System.out.println("Sorted Employee Names (Case-Insensitive):");
          System.out.println(Arrays.toString(employeeNames));
    }
```

```
}
```