OOPJ CCEE Practice Quiz

Total points 11/20

The respondent's email (**chitransh1709@gmail.com**) was recorded on submission of this form.

0 of 0 points

PRN (12 Digit) *

240840320030

Name *

Chitransh Mrigank Singh

Center *

Kharghar

Questions

11 of 20 points

✓ Which method in the RandomAccessFile class is used to move the file pointer to a specific position? * 1/1

- [✓ a) seek()] ✓
- b) move()
- c) locate()
- d) find()

✓ Which collection class is suitable if your application requires fast random access but infrequent insertions and deletions? * 1/1

- a) LinkedList
- [✓ b) ArrayList] ✓
- c) HashSet
- d) PriorityQueue

✗ Given the following code snippet: * 0/1

```
Map<Integer, String> map = new TreeMap<>();
map.put(1, "A");
map.put(2, "B");
map.put(null, "C");
```

What will happen when this code is executed?

- a) The code will compile and run normally.
- b) The code will throw a NullPointerException at runtime.
- c) The code will throw a ClassCastException.
- [✓ d) The code will throw a IllegalArgumentException.] ✗

Correct answer

- [✓ b) The code will throw a NullPointerException at runtime.]

✓ How do you retrieve all keys from a Map in Java? * 1/1

```
Map<String, Integer> map = new HashMap<>();
map.put("A", 1);
map.put("B", 2);
map.put("C", 3);
```

a) map.getKeys()

b) map.values()

[✓ c) map.keySet()] ✓

d) map.entrySet()

✗ Consider the following code snippet: * 0/1

```
public class CustomException extends Exception {}

public class Test {
    public static void main(String[] args) {
        try {
            throw new CustomException();
        } catch (Exception e) {
            throw e;
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

a) It will print "Finally block executed" and throw CustomException.

[✓ b) It will print "Finally block executed" and terminate normally.] ✗

c) It will result in a compilation error.

d) It will print nothing and terminate normally.

Correct answer

● c) It will result in a compilation error.

✓ What is the output of the following code snippet? * 1/1

```
List<Integer> list = new LinkedList<>(Arrays.asList(1, 2, 3, 4));
list.add(0, 5);
list.add(5, 6);
System.out.println(list);
```

[✓ a) [5, 1, 2, 3, 4, 6]] ✓

b) [1, 2, 3, 4, 5, 6]

c) [5, 1, 2, 3, 4]

d) IndexOutOfBoundsException

✓ What will happen if you use the following code and the map contains duplicate values? *1/1

```
Map<Integer, String> map = new HashMap<>();
map.put(1, "apple");
map.put(2, "banana");
map.put(3, "apple");
Set<String> set = new HashSet<>(map.values());
System.out.println(set);
```

a) It will print all the values: [apple, banana, apple].

[✓ b) It will print only unique values: [apple, banana].] ✓

c) It will throw a ConcurrentModificationException.

d) It will remove duplicate keys from the map.

✗ Which of the following is the most efficient collection type to use when frequent insertions and deletions occur at both ends of a list? *0/1

a) ArrayList

b) LinkedList

c) Vector

[✓ d) PriorityQueue] ✗

Correct answer

[✓ b) LinkedList]

✓ What would happen in the following scenario? * 1/1

```
Set<String> set = new HashSet<>();
set.add("one");
set.add(null);
set.add("two");
set.add(null);
System.out.println(set.size());
```

- a) 2
- [✓ b) 3] ✓
- c) NullPointerException
- d) Compilation Error

✓ Which of the following correctly describes the difference between HashMap and Hashtable? * 1/1

- a) HashMap is synchronized, whereas Hashtable is not.
- [✓ b) HashMap allows null keys and values, whereas Hashtable does not.] ✓
- c) Both HashMap and Hashtable allow null keys.
- d) Hashtable is more efficient than HashMap.

✗ What is the purpose of the WeakHashMap in Java? * 0/1

- a) To allow keys to be garbage-collected when no longer referenced.
- b) To improve performance over HashMap.
- [✓ c) To ensure thread safety.] ✗
- d) To enforce unique values.

Correct answer

[✓ a) To allow keys to be garbage-collected when no longer referenced.]

✗ What happens if a catch block is defined for a checked exception but that exception is not thrown within the try block? * 0/1

- [✓ a) Compile-time error.] ✗
- b) Runtime exception.
- c) The catch block will be ignored.
- d) The program will not compile if no catch block matches.

Correct answer

[✓ c) The catch block will be ignored.]

✗ What will be the output if you run the program? * 0/1

```
try {
   throw new Exception("Test Exception");
} finally {
   throw new RuntimeException("Runtime Exception in finally");
}
```

- a) The program will compile successfully but throw an Exception.
- b) The program will compile successfully but throw a RuntimeException.
- [✓ c) The program will not compile due to the unchecked exception in the finally block.] ✗
- d) The program will compile successfully but throw both Exception and RuntimeException.

Correct answer

● b) The program will compile successfully but throw a RuntimeException.

✓ What happens if you attempt to modify a collection while iterating over it using an Iterator? *1/1

● a) It throws a ConcurrentModificationException. ✓

○ b) It modifies the collection without issues.

○ c) It creates an infinite loop.

○ d) It modifies only elements after the iterator's current position.

✗ What happens when you use the following code snippet and the file already exists? *0/1

```
FileOutputStream fos = new FileOutputStream("test.txt", false);
```

○ a) It will throw an exception.

● b) It will append to the file. ✗

○ c) It will overwrite the file.

○ d) It will open the file in read-only mode.

Correct answer

● c) It will overwrite the file.

✗ Consider the following code: * 0/1

```
List<String> list = new ArrayList<>();
list.add("one");
list.add("two");
list.add("three");
List<String> sublist = list.subList(1, 2);
sublist.add("four");
System.out.println(list);
```

What will be printed?

● a) [one, two, four] ✗

○ b) [one, two, four, three]

○ c) [one, two, three]

○ d) ConcurrentModificationException

Correct answer

● b) [one, two, four, three]

✓ Which of the following best describes the term "exception chaining" in Java? *1/1

● a) Wrapping one exception inside another. ✓

○ b) Catching multiple exceptions in a single catch block.

○ c) Nesting try blocks.

○ d) Handling checked exceptions using unchecked exceptions.

✗ Given the following code, what will happen if the file does not exist? *0/1
BufferedReader br = new BufferedReader(new FileReader("existingFile.txt"));

● a) It will create the file if it does not exist. ✗

○ b) It will throw a FileNotFoundException.

○ c) It will return null.

d) It will return an empty string.

Correct answer

[✓ b) It will throw a FileNotFoundException.]

✓ What is the output of the following code? * 1/1

```
List<String> list = Arrays.asList("apple", "banana", "cherry");
ListIterator<String> iterator = list.listIterator();
while (iterator.hasNext()) {
    System.out.print(iterator.next() + " ");
    if (iterator.nextIndex() == 2) {
        iterator.previous();
    }
}
```

a) apple banana banana cherry

b) apple banana cherry

c) IndexOutOfBoundsException

[✓ d) Infinite Loop] ✓

✓ Which of the following options guarantees insertion-order preservation but with no duplicates? * 1/1

a) TreeSet

b) HashSet

[✓ c) LinkedHashSet] ✓

d) ArrayList