

Java Full Stack Developer

Candidate-ID	0012500078
Candidate Name	siddhardha kireeti
Email	siddhardhadepu@gmail.com
Mobile	8142268525
Source	Careers

Screening Test

Completed on : 15-Nov-2025 07:23 AM

#	Question	Response	Marks
1	Do you have professional experience between 1 to 5 years?	Yes	1
2	Would you be open to relocating to Hyderabad?	Yes	1
3	Are you familiar with Agile methodologies, including Scrum or Kanban?	Yes	1
4	Are you open to traveling for work when necessary, either domestically or internationally?	Yes	1
5	Will you be flexible any role in Software Engineering	Yes	1
6	Explain about yourself why you are the right candidate for this Job	video response	1
7	Have you recently completed your education and training, and there no is major Gap (more then one year after the education)	Yes	1
8	Are you, willing to work on-premises (there is no work from home)	Yes	1
9	Are you, available immediately to join us, if we hire you	Yes	1
10	if we hire you, will you be prepared to stay and work with us at least for two years	Yes	1

Overall Score : 10/10 (Promoted)

Coding Test

Completed on : 15-Nov-2025 01:06 PM

1. Write a program or method to find the smallest and largest numbers in an array of integers. Implement two methods: `findSmallest(int[] numbers)`: Returns the smallest number in the array. `findLargest(int[] numbers)`: Returns the largest number in the array. Sample Input : If the input array is null or empty, the method should throw an `IllegalArgumentException` with the message "Array cannot be null or empty". Use Java's Stream API to solve the problem. Expected Output : Smallest number: 1 Largest number: 20

Marks Scored : 2

```
public class NumberUtils {

    public static int findSmallest(int[] numbers) {
        if (numbers == null || numbers.length == 0) {
            throw new IllegalArgumentException("Array cannot be null or empty");
        }
        // Implement the logic here
        int smallest = numbers[0];
        for (int i = 1; i < numbers.length; i++) {
            if (numbers[i] < smallest) {
                smallest = numbers[i];
            }
        }
        return smallest;
    }

    public static int findLargest(int[] numbers) {
        if (numbers == null || numbers.length == 0) {
            throw new IllegalArgumentException("Array cannot be null or empty");
        }
        // Implement the logic here
        int largest = numbers[0];
        for (int i = 1; i < numbers.length; i++) {
            if (numbers[i] > largest) {
                largest = numbers[i];
            }
        }
        return largest;
    }

    public static void main (String[] args) {
        int[] testArray1 = {3, 7, 2, 9, 1, 5};
        System.out.println("Smallest" + findSmallest(testArray1));
        System.out.println("Largest" + findLargest(testArray1));

        int[] testArray2 = {-5, -2, -8, -1};
        System.out.println("Smallest" + findSmallest(testArray2));
        System.out.println("Largest" + findLargest(testArray2));

        int[] testArray3 = {42};
    }
}
```

```

System.out.println("Smallest" + findSmallest(testArray3));
System.out.println("Largest" + findLargest(testArray3));
}
try {
    int[] emptyArray = {};
    findSmallest(emptyArray);
} catch (IllegalArgumentException e) {
    System.out.println("Correctly handled empty array: " + e.getMessage());
}
}
}
}

```

2. Question: Unique Words Extractor Write a program to extract all unique words from a given text. A word is defined as a sequence of alphanumeric characters separated by non-word characters (e.g., spaces, punctuation). The program should meet the following criteria: Extract words that are at least 8 characters long. Convert all words to lowercase for uniqueness. Preserve the order of their first appearance in the text. Implement the functionality in the extractUniqueWords method. The method should return a Set containing the filtered words. Sample Input : "The development of artificial intelligence is advancing rapidly. Innovative technologies are shaping the future of the world." Expected Output : Unique words with at least 8 characters: development artificial intelligence advancing innovative technologies

Marks Scored : 4

```

import java.util.*;
import java.util.stream.Collectors;
import java.util.regex.Pattern;

public class WordProcessor {

    public static Set extractUniqueWords(String content) {

        if (content == null || content.trim().isEmpty()) {
            return new LinkedHashSet<>();
        }
        Pattern pattern = Pattern.compile("\\W+");
        String[] words = pattern.split(content);

        Set uniqueWords = new LinkedHashSet<>();

        for (String word : words) {
            if (word.length() >= 8) {
                uniqueWords.add(word.toLowerCase());
            }
        }
        return uniqueWords;
    }

    public static void main(String[] args) {

```

```
// Example content for testing
String content = "The development of artificial intelligence is advancing rapidly. " +
    "Innovative technologies are shaping the future of the world.";

// Extract unique words with at least 8 characters
Set uniqueWords = extractUniqueWords(content);

// Print the unique words
System.out.println("Unique words with at least 8 characters:");
uniqueWords.forEach(System.out::println);
}
}
```

3. Question Write a program to demonstrate most simple usage, application of design pattern prototype. Expected Output: Original: Shape[type=Circle, color=Red] Cloned: Shape[type=Circle, color=Red] After modifying cloned object: Original: Shape[type=Circle, color=Red] Cloned: Shape[type=Circle, color=Blue]

Marks Scored : 0

```
// PrototypeDemo.java
class Shape implements Cloneable {
    private String type;
    private String color;

    public Shape(String type, String color) {
        // CODE HERE
        this.type = type;
        this.color = color;
    }

    public String getType() {
        // CODE HERE
        return type;
    }

    private String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    @Override
    protected Shape clone() {
        try {
            return (Shape) super.clone();
        } catch (CloneNotSupportedException e) {
            throw new RuntimeException("Clone not supported" , e);
        }
    }
}
```

```
    }  
}  
  
@Override  
public String toString () {  
    return "Shape[type =" + type +" , color = "+color +"]";  
}  
}  
  
public class PrototypeDemo {  
    public static void main(String[] args) {  
        Shape original = new Shape("Circle", "Red" );  
        Shape cloned = original.clone();  
  
        System.out.println("Original" + original);  
        System.out.println("Cloned" + cloned);  
  
        cloned.setColor("Blue");  
  
        System.out.println("After modifying cloned object");  
        System.out.println("Original:" + original);  
        System.out.println("Cloned" + cloned);  
  
    }  
}
```

Overall Score : 6/12 (Failed)