

## **Sheet Programming 3**

Q1. Which of the following errors occurs during the compilation phase and prevents bytecode generation?

- a) Logical Errors
- b) Runtime Errors
- c) Compile-time Errors
- d) JVM Errors

Q2. Logical errors in Java:

- a) Are reported by the compiler
- b) Cause the JVM to crash
- c) Produce incorrect results without any error message
- d) Are the easiest to fix

Q3. Runtime errors in Java are represented by:

- a) Checked exceptions only
- b) Objects of class Exception
- c) Objects of class Error
- d) Objects of class Throwable only

Q4. All errors and exceptions in Java inherit from:

- a) java.lang.Exception

b) java.lang.Error

c) java.lang.Throwable

d) java.lang.RuntimeException

Q5. Which of the following is non-recoverable and generally not handled by applications?

a) IOException

b) SQLException

c) OutOfMemoryError

d) NullPointerException

Q6. Which of the following is a checked exception?

a) ArithmeticException

b) IOException

c) NullPointerException

d) ArrayIndexOutOfBoundsException

Q7. Which feature differentiates checked exceptions from unchecked exceptions?

a) Checked exceptions are runtime only

b) Checked exceptions must be handled or declared

c) Unchecked exceptions are always recoverable

d) Checked exceptions extend RuntimeException

Q8. Which exception occurs when trying to divide an integer by zero?

a) NullPointerException

b) ArithmeticException

c) NumberFormatException

d) ClassCastException

Q9. What exception will this code throw? String s=null; s.length();

a) ArrayIndexOutOfBoundsException

b) ClassCastException

c) NullPointerException

d) NumberFormatException

Q10. Which exception occurs when trying to cast an Integer object to a String?

a) ArithmeticException

b) ClassCastException

c) NumberFormatException

d) IllegalArgumentException

Q11. Which Java class represents a file or directory but does not read/write file contents?

A) FileInputStream

B) File

C) Scanner

D) PrintWriter

Q12. What does File.exists() return when the path points to an existing file?

A) false

B) true

C) Throws exception

D) null

Q13. Which class is typically used to write formatted text to a file?

A) Scanner

B) FileOutputStream

C) PrintWriter

D) DataOutputStream

Q14. Which PrintWriter method adds a newline after the output?

A) print()

B) printf()

C) println()

D) write()

Q15. Which Scanner method reads an entire line including spaces?

A) next()

B) nextInt()

C) nextLine()

D) nextDouble()

Q16. Default delimiter used by Scanner is:

A) Comma ,

B) Newline only

C) Whitespace

D) Semicolon ;

Q17. Which exception is unchecked (runtime)?

- A) IOException
- B) SQLException
- C) NullPointerException
- D) FileNotFoundException

Q18. File.isDirectory() returns true when path is a:

- A) Regular file
- B) Directory
- C) Nonexistent path
- D) Hidden file only

Q19. Which method of File returns the file name?

- A) getAbsolutePath()
- B) getParent()
- C) getName()
- D) getPath()

Q20. PrintWriter when created with a filename that does not exist will:

- A) Throw FileNotFoundException
- B) Create the file
- C) Return null
- D) Create a directory

Q21. Which stream is for reading raw bytes?

A) FileReader

B) FileInputStream

C) BufferedReader

D) PrintWriter

Q22. File.length() returns:

A) Number of characters

B) File size in bytes

C) Number of lines

D) null

Q23. Scanner.nextInt() when token is not an integer will throw:

A) InputMismatchException

B) IOException

C) NumberFormatException

D) NullPointerException

Q24. Which class writes primitive types in binary form?

A) DataOutputStream

B) PrintWriter

C) FileWriter

D) BufferedReader

Q25. Which File method deletes a file?

A) remove()

B) delete()

C) erase()

D) unlink()

Q26. To test whether a path is absolute, use:

A) isFile()

B) isAbsolute()

C) getAbsolutePath()

D) exists()

Q27. Which File method returns last modification time?

A) lastModified()

B) getLastUpdate()

C) getModified()

D) modified()

Q28. Which escape sequence represents a newline?

A) \t

B) \\\

C) \n

D) \"

Q29. Which class method checks whether a character is a digit?

A) Character.isLetter()

B) Character.isDigit()

C) Character.toUpperCase()

D) Character.getNumeric()

Q30. Strings in Java are:

A) mutable objects

B) primitive types

C) immutable objects

D) synchronized by default

Q31. Which String method returns number of characters?

A) size()

B) length()

C) count()

D) charCount()

Q32. What does s.charAt(0) return for "Hi"?

A) 'H'

B) "H"

C) 0

D) null

Q33. Which method checks full-string regex match?

A) contains()

B) matches()



C) search()

D) find()

Q34. Which String method removes spaces?

A) strip()

B) trim()

C) chomp()

D) remove()

Q35. Which method concatenates two strings?

A) append()

B) concat()

C) +=

D) push()

Q36. equalsIgnoreCase() compares strings:

A) Case-sensitive

B) Case-insensitive

C) By length

D) By reference

Q37. indexOf() returns:

A) last occurrence index

B) first occurrence index

C) number of occurrences

D) boolean

Q38. Which method returns char array?

A) toBytes()

B) chars()

C) toCharArray()

D) asChars()

Q39. Which class provides mutable sequence?

A) String

B) StringBuilder

C) Character

D) byte[]

Q40. StringBuilder is not synchronized. That makes it:

A) slower

B) faster

C) deprecated

D) immutable

Q41. Which method reverses sequence?

A) flip()

B) reverse()

C) invert()

D) reversed()

Q42. StringTokenizer by default splits on:

- A) commas
- B) whitespace
- C) semicolons
- D) punctuation

Q43. Which method returns number of tokens?

- A) size()
- B) countTokens()
- C) tokensLeft()
- D) length()

Q44. replace(char old, char new) replaces:

- A) first only
- B) all occurrences
- C) regex matches
- D) nothing

Q45. toUpperCase() returns:

- A) Mutated original
- B) New uppercase string
- C) char[]
- D) void

Q46. substring(begin,end) includes:

- A) Both
- B) begin inclusive, end exclusive
- C) begin exclusive
- D) Neither

Q47. Which package contains regex?

- a) java.io.regex
- b) java.util.regex
- c) java.lang.regex
- d) java.text.regex

Q48. What does \d match?

- a) Any letter
- b) Any digit
- c) Any whitespace
- d) Any special character

Q49. Which class compiles regex?

- a) Matcher
- b) Regex
- c) Pattern
- d) Scanner

Q50. Which Matcher method checks full match?

- a) find()
- b) lookingAt()
- c) matches()
- d) search()

Q51. What does [a-z]+ match?

- a) One or more lowercase letters
- b) One lowercase letter
- c) Any letter or digit
- d) Only 'abc'

Q52. In JavaFX, main class extends:

- a) Stage
- b) Scene
- c) Application
- d) Node

Q53. Top-level JavaFX container:

- a) Scene
- b) Pane
- c) Stage
- d) Node

Q54. Which method starts JavaFX app?

- a) main()

b) init()

c) start()

d) run()

Q55. What does \w match?

a) Whitespace

b) Word chars

c) Non-word chars

d) Digits

Q56. Which displays static text?

a) TextField

b) TextArea

c) Label

d) Button

Q57. Meaning of \* in regex?

a) Exactly one

b) Zero or more

c) One or more

d) Optional

Q58. Which allows multi-line input?

a) Label

b) TextField

c) PasswordField

d) TextArea

Q59. How to display JavaFX window?

a) setVisible

b) open

c) show

d) display

Q60. What does ^ mean in regex?

a) End

b) Start

c) Any char

d) Escape

Q61. Which class creates matcher?

a) String

b) Pattern

c) Matcher

d) RegexEngine

Q62. What does \$ represent in regex?

a) Start

b) End

c) Digit

d) Wildcard

Q63. Which returns matched substring?

a) text()

b) value()

c) group()

d) match()

Q64. What does \s+ match?

a) Digits

b) Letters

c) Whitespace

d) Special chars

Q65. Which is correct about Strings?

A) mutable

B) immutable

C) mutable if final

D) mutable with concat

Q66. Best for efficient string manipulation?

A) String

B) StringBuilder

C) StringBuffer

D) StringTokenizer



Q67. Regex for exactly 5 digits?

A) `\d+`

B) `\d{5}`

C) `[0-9]*`

D) `\d{1,5}`

Q68. Output: `'Hello World'.indexOf("o")`

A) 4

B) 5

C) 6

D) -1

Q69. Regex matching both 'color' and 'colour'?

A) `colou+r`

B) `colou?r`

C) `colo+r`

D) `col[o,u]r`

## Essay Questions (Tasks)

- Task 1: Write a Java program that appends 'x' 10,000 times using String and StringBuilder, compare performance.

```
> public class Task1 {  
>     public static void main(String[] args) {  
        long start1 = System.currentTimeMillis();  
        String s = "";  
        for (int i = 0; i < 10000; i++) {  
            s += "x";  
        }  
        long end1 = System.currentTimeMillis();  
        System.out.println("String (+) time: " + (end1 - start1) + " ms");  
  
        long start2 = System.currentTimeMillis();  
        StringBuilder sb = new StringBuilder();  
        for (int i = 0; i < 10000; i++) {  
            sb.append("x");  
        }  
        long end2 = System.currentTimeMillis();  
        System.out.println("StringBuilder time: " + (end2 - start2) + " ms");  
    }  
}
```

- Task 2: Split 'Java is fun and powerful!' using split("\\s+") and print words.

```
> public class Task2 {  
>     public static void main(String[] args) {  
        String text = "Java is fun and powerful!";  
        String[] words = text.split(regex: "\\s+");  
  
        for (String w : words) {  
            System.out.println(w);  
        }  
    }  
}
```

- Task 3: Write a regex to validate a basic email address; test examples.

```
> public class Task3 {  
>     public static void main(String[] args) {  
        String regex = "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$";  
  
        String[] tests = {  
            "student@mail.com",  
            "student@mail"  
        };  
  
        for (String email : tests) {  
            System.out.println(email + " -> " + email.matches(regex));  
        }  
    }  
}
```

- Task 4: Extract all hashtags from: 'Learning #Java is great! #Coding #Regex'

```
1 import java.util.regex.*;  
2  
3 public class Task4 {  
4     public static void main(String[] args) {  
5         String post = "Learning #Java is great! #Coding #Regex";  
6  
7         Pattern p = Pattern.compile(regex: "#\\w+");  
8         Matcher m = p.matcher(post);  
9  
10        while (m.find()) {  
11            System.out.println(m.group());  
12        }  
13    }  
14 }  
15
```

- Task 5: Capture Year, Month, Day from '2025-10-03' using regex groups.

```
import java.util.regex.*;

public class Task5 {
    public static void main(String[] args) {
        String date = "2025-10-03";

        Pattern p = Pattern.compile(regex: "(\\d{4})-(\\d{2})-(\\d{2})");
        Matcher m = p.matcher(date);

        if (m.find()) {
            System.out.println("Year: " + m.group(1));
            System.out.println("Month: " + m.group(2));
            System.out.println("Day: " + m.group(3));
        }
    }
}
```