Class 10

Chapter – 16

String Handling

A. TICK (✓) THE CORRECT ANSWER

Q1. Which operator is used to concatenate strings?

a. *
b. ^
c. +
d. #

✅ **Answer:** c. +

❗ **Explanation:**
In Java, + joins (concatenates) strings.
Example: "Hello" + "World" → "HelloWorld"

Q2. Output of code

```
String s = "ORANGE";
String s1 = "education";
System.out.println(s + s1.toUpperCase());
```

a. ORANGEEDUCATION
b. ORANGEeducation
c. ORANGE EDUCATION
d. None

✅ **Answer:** a. ORANGEEDUCATION

❗ **Explanation:**

- toUpperCase() → EDUCATION
- Concatenation → ORANGE + EDUCATION

Q3. Method to compare two strings

a. equals()
b. Equals()
c. isequal()
d. All

✅ **Answer:** a. equals()

💡 **Explanation:**
Java is **case-sensitive** → only equals() is correct.

Q4. Output of code

```
String Str1 = "I am a good student";
String Str2 = "am";
String Str3 = "was";
System.out.println(Str1.replace(Str2, Str3));
```

a. I was a good student
b. I am a good student
c. I was good student
d. Nothing

✅ **Answer:** a. I was a good student

💡 **Explanation:**
replace("am","was") → replaces only "am"

Q5. Correct way to declare String

a. String str = new String("INDIA");
b. String str = "INDIA";
c. Both a and b
d. None

✅ **Answer:** c. Both a and b

💡 **Explanation:**
Both are valid ways.

Q6. Output

```
String s = "Computer";
s = s.concat(" Applications");
System.out.println(s);
```

a. Applications
b. ComputerApplications
c. Computer Applications
d. Compilation error

✅ **Answer:** c. Computer Applications

Explanation:
Space is included in " Applications"

Q7. Method to convert string to uppercase

a. ToUpper()
b. toUppercase()
c. toUpperCase()
d. convertToUpper()

Answer: c. toUpperCase()

Q8. Output
String s1 = "aaa";
System.out.println(s1.indexOf('a'));

a. 0
b. 1
c. 2
d. 3

Answer: a. 0

Explanation:
Index starts from 0 → first 'a' at position 0.

Q9. Output
String s = "Hello World";
System.out.println(s.substring(1,7).lastIndexOf('o'));

a. 0
b. -1
c. 1
d. 3

Answer: d. 3

Explanation:
substring → "ello W"
Last 'o' → index 3

Q10. Remove spaces method

a. trim()
b. strip()

c. removeSpaces()
d. clean()

✅ **Answer:** a. trim()

✅ B. Fill in the blanks

1. A **String** is a sequence of characters.
2. Combining two strings is called **concatenation**
3. Return type of replace() is **String**
4. trim() removes **leading and trailing spaces**
5. Method to convert primitive to string → **valueOf()**

C. Answer the Following Questions (Programs)

◆ Q1 (i) Count words starting with 'A'

```
class Q1_i
{
   public static void main(String args[])
   {
      String s = "An apple and a mango";
      String words[] = s.split(" ");
      int count = 0;

      for(int i=0; i<words.length; i++)
      {
         if(words[i].startsWith("A") || words[i].startsWith("a"))
         {
            count++;
         }
      }

      System.out.println("Words starting with A: " + count);
   }
}
```

◆ Q1 (ii) Count vowels and consonants

```
class Q1_ii
{
   public static void main(String args[])
   {
      String s = "Computer Applications";
      int vowels = 0, consonants = 0;

      for(int i=0; i<s.length(); i++)
      {
```

```
            char ch = s.charAt(i);

            if(Character.isLetter(ch))
            {
               if("AEIOUaeiou".indexOf(ch) != -1)
                  vowels++;
               else
                  consonants++;
            }
         }

         System.out.println("Vowels: " + vowels);
         System.out.println("Consonants: " + consonants);
      }
   }
```

◆ Q2 Replace character and count

```
class Q2
{
   public static void main(String args[])
   {
      String s = "INDIA IS MY COUNTRY";
      char ch = 'I';
      int count = 0;

      for(int i=0; i<s.length(); i++)
      {
         if(s.charAt(i) == ch)
            count++;
      }

      String newStr = s.replace(ch, '*');

      System.out.println("Count of " + ch + ": " + count);
      System.out.println("New String: " + newStr);
   }
}
```

◆ Q3 Longest word in string

```
class Q3
{
   public static void main(String args[])
   {
      String s = "Computer Applications deals with application";
      String words[] = s.split(" ");

      String longest = "";

      for(int i=0; i<words.length; i++)
      {
         if(words[i].length() > longest.length())
            longest = words[i];
      }
```

```
        System.out.println("Longest word: " + longest);
    }
}
```

◆ Q4 Smallest word

```
class Q4
{
    public static void main(String args[])
    {
        String s = "Computer Applications deals with application";
        String words[] = s.split(" ");

        String smallest = words[0];

        for(int i=0; i<words.length; i++)
        {
            if(words[i].length() < smallest.length())
                smallest = words[i];
        }

        System.out.println("Smallest word: " + smallest);
    }
}
```

◆ Q5 Convert to palindrome

```
class Q5
{
    public static void main(String args[])
    {
        String s = "general";
        String rev = "";

        for(int i=s.length()-1; i>=0; i--)
        {
            rev = rev + s.charAt(i);
        }

        System.out.println("Palindrome word: " + s + rev);
    }
}
```

◆ Q6 ASCII values

```
class Q6
{
    public static void main(String args[])
    {
        String s = "ABCDE";
```

```
        for(int i=0; i<s.length(); i++)
        {
            char ch = s.charAt(i);
            System.out.println(ch + " = " + (int)ch);
        }
    }
}
```

◆ Q7 Abbreviated name

```
class Q7
{
    public static void main(String args[])
    {
        String name = "Ram Kumar Sharma";
        String words[] = name.split(" ");

        System.out.println(words[2] + ", " + words[0].charAt(0) + "." + words[1].charAt(0) + ".");
    }
}
```

◆ Q8 Frequency of characters

```
class Q8
{
    public static void main(String args[])
    {
        String s = "India New Delhi".toUpperCase();

        for(char ch='A'; ch<='Z'; ch++)
        {
            int count = 0;

            for(int i=0; i<s.length(); i++)
            {
                if(s.charAt(i) == ch)
                    count++;
            }

            if(count > 0)
                System.out.println(ch + " = " + count);
        }
    }
}
```

◆ Q9 Word with maximum vowels

```
class Q9
{
    public static void main(String args[])
    {
        String s = "I AM STUDYING COMPUTER APPLICATIONS";
```

```
        String words[] = s.split(" ");

        String result = "";
        int max = 0;

        for(int i=0; i<words.length; i++)
        {
            int count = 0;

            for(int j=0; j<words[i].length(); j++)
            {
                char ch = words[i].charAt(j);
                if("AEIOU".indexOf(ch) != -1)
                    count++;
            }

            if(count > max)
            {
                max = count;
                result = words[i];
            }
        }

        System.out.println("Word: " + result);
    }
}
```

◆ Q10 Reverse each word

```
class Q10
{
    public static void main(String args[])
    {
        String s = "India is great";
        String words[] = s.split(" ");

        for(int i=0; i<words.length; i++)
        {
            String rev = "";

            for(int j=words[i].length()-1; j>=0; j--)
            {
                rev = rev + words[i].charAt(j);
            }

            System.out.print(rev + " ");
        }
    }
}
```

◆ Q11 Replace word

```
class Q11
{
    public static void main(String args[])
```

```
    {
        String s = "Every person had a star, every star had a friend";

        String newStr = s.replace("star", "moon");

        System.out.println(newStr);
    }
}
```

◆ Q12 Unique word

```
class Q12
{
    public static void main(String args[])
    {
        String s = "sky";
        boolean unique = true;

        for(int i=0; i<s.length(); i++)
        {
            for(int j=i+1; j<s.length(); j++)
            {
                if(s.charAt(i) == s.charAt(j))
                {
                    unique = false;
                    break;
                }
            }
        }

        if(unique)
            System.out.println("Unique word");
        else
            System.out.println("Not unique");
    }
}
```

◆ Q13 Replace vowels with next letter

```
class Q13
{
    public static void main(String args[])
    {
        String s = "Kolkata";
        String result = "";

        for(int i=0; i<s.length(); i++)
        {
            char ch = s.charAt(i);

            if("AEIOUaeiou".indexOf(ch) != -1)
                result = result + (char)(ch + 1);
            else
                result = result + ch;
        }
```

```
        System.out.println(result);
    }
}
```

◆ Q14 Remove vowels and digits

```
class Q14
{
    public static void main(String args[])
    {
        String s = "Pincode of Sodepur is Kolkata - 700115";
        String result = "";

        for(int i=0; i<s.length(); i++)
        {
            char ch = s.charAt(i);

            if("AEIOUaeiou0123456789".indexOf(ch) == -1)
                result = result + ch;
        }

        System.out.println(result);
    }
}
```

◆ Q15 Replace word

```
class Q15
{
    public static void main(String args[])
    {
        String s = "India is my country";

        System.out.println(s.replace("my", "our"));
    }
}
```

◆ Q16 Repeated words

```
class Q16
{
    public static void main(String args[])
    {
        String s = "NEXT TIME THERE WON'T BE A NEXT TIME";
        String words[] = s.split(" ");

        for(int i=0; i<words.length; i++)
        {
            int count = 0;

            for(int j=0; j<words.length; j++)
```

```
            {
                if(words[i].equals(words[j]))
                    count++;
            }

            if(count > 1)
                System.out.println(words[i]);
        }
    }
}
```

◆ Q17 Smallest & largest palindrome

```
class Q17
{
    public static void main(String args[])
    {
        String s = "MOM AND MADAM BOTH CAN SPEAK MALAYALAM";
        String words[] = s.split(" ");

        String small = "", large = "";

        for(int i=0; i<words.length; i++)
        {
            String w = words[i];
            String rev = "";

            for(int j=w.length()-1; j>=0; j--)
                rev = rev + w.charAt(j);

            if(w.equals(rev))
            {
                if(small.equals("") || w.length() < small.length())
                    small = w;

                if(w.length() > large.length())
                    large = w;
            }
        }

        System.out.println("Smallest: " + small);
        System.out.println("Largest: " + large);
    }
}
```

◆ Q18 Words starting and ending with same letter

```
class Q18
{
    public static void main(String args[])
    {
        String s = "level madam apple radar cat";
        String words[] = s.split(" ");

        for(int i=0; i<words.length; i++)
```

```
        {
            String w = words[i];

            if(w.charAt(0) == w.charAt(w.length()-1))
            {
                System.out.println(w);
            }
        }
    }
}
```

◆ Q19 Cities: consonant start & vowel end

```
class Q19
{
    public static void main(String args[])
    {
        String cities[] = {"DELHI","KOLKATA","JAMSHEDPUR","LUCKNOW","PATNA"};

        for(int i=0; i<cities.length; i++)
        {
            String c = cities[i];

            char first = c.charAt(0);
            char last = c.charAt(c.length()-1);

            if("AEIOU".indexOf(first) == -1 && "AEIOU".indexOf(last) != -1)
            {
                System.out.println(c);
            }
        }
    }
}
```

◆ Q20 Create single-dimensional array of size 10

```
class Q20
{
    public static void main(String args[])
    {
        int arr[] = new int[10];

        for(int i=0; i<10; i++)
        {
            arr[i] = i+1;
        }

        for(int i=0; i<10; i++)
        {
            System.out.print(arr[i] + " ");
        }
    }
}
```

◆ Q21 Names in alphabetical order

```
class Q21
{
   public static void main(String args[])
   {
      String names[] = {"Ram","Amit","Sita","John","Ravi"};

      for(int i=0; i<names.length-1; i++)
      {
         for(int j=i+1; j<names.length; j++)
         {
            if(names[i].compareTo(names[j]) > 0)
            {
               String temp = names[i];
               names[i] = names[j];
               names[j] = temp;
            }
         }
      }

      System.out.println("Sorted Names:");
      for(int i=0; i<names.length; i++)
      {
         System.out.println(names[i]);
      }
   }
}
```

◆ Q21 Pattern (JAVA)

(a)

```
JAVA
JAV
JA
J
```

```
class Q21_a
{
   public static void main(String args[])
   {
      String s = "JAVA";

      for(int i=s.length(); i>0; i--)
      {
         System.out.println(s.substring(0,i));
      }
   }
}
```

(b)

```
J
JA
JAV
JAVA
```

```
class Q21_b
{
```

```
    public static void main(String args[])
    {
        String s = "JAVA";

        for(int i=1; i<=s.length(); i++)
        {
            System.out.println(s.substring(0,i));
        }
    }
}
```

(c)

```
JAVA
AVA
VA
A
```

```
class Q21_c
{
    public static void main(String args[])
    {
        String s = "JAVA";

        for(int i=0; i<s.length(); i++)
        {
            System.out.println(s.substring(i));
        }
    }
}
```

(d)

```
A
AV
AVA
AVAJ
```

```
class Q21_d
{
    public static void main(String args[])
    {
        String s = "JAVA";
        String rev = "";

        for(int i=s.length()-1; i>=0; i--)
        {
            rev = rev + s.charAt(i);
            System.out.println(rev);
        }
    }
}
```

◆ Q22 Alphabet patterns

(a)

```
class Q22_a
{
    public static void main(String args[])
    {
```

```
        for(char ch='E'; ch>='A'; ch--)
        {
            for(char c='A'; c<=ch; c++)
            {
                System.out.print(c);
            }
            System.out.println();
        }
    }
}
```

(b)

```
class Q22_b
{
    public static void main(String args[])
    {
        for(int i=0; i<5; i++)
        {
            for(char c=(char)('A'+i); c<='E'; c++)
            {
                System.out.print(c);
            }
            System.out.println();
        }
    }
}
```

(c)

```
class Q22_c
{
    public static void main(String args[])
    {
        for(char ch='A'; ch<='E'; ch++)
        {
            for(char c='A'; c<=ch; c++)
            {
                System.out.print(c);
            }
            System.out.println();
        }
    }
}
```

(d)

```
class Q22_d
{
    public static void main(String args[])
    {
        char start='A';

        for(int i=1; i<=4; i++)
        {
            char ch=start;

            for(int j=1; j<=i; j++)
            {
                System.out.print(ch);
                ch++;
            }
```

```
            start+=2;
            System.out.println();
        }
    }
}
```

◆ Q23 Selection sort (names + marks)

```
class Q23
{
    public static void main(String args[])
    {
        String name[] = {"Ram","Amit","Sita","John","Ravi"};
        int marks[] = {80,90,75,85,88};

        for(int i=0; i<name.length-1; i++)
        {
            int min=i;

            for(int j=i+1; j<name.length; j++)
            {
                if(name[j].compareTo(name[min]) < 0)
                    min=j;
            }

            String tempName=name[i];
            name[i]=name[min];
            name[min]=tempName;

            int tempMarks=marks[i];
            marks[i]=marks[min];
            marks[min]=tempMarks;
        }

        for(int i=0; i<name.length; i++)
        {
            System.out.println(name[i]+" "+marks[i]);
        }
    }
}
```

◆ Q24 Menu-driven PROGRAM

```
import java.util.*;

class Q24
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        String s = "PROGRAM";

        System.out.println("Enter choice:");
        int ch = sc.nextInt();
```

```
        if(ch==1)
        {
            for(int i=0; i<s.length(); i++)
            {
                System.out.println(s.substring(i));
            }
        }
        else if(ch==2)
        {
            for(int i=1; i<=s.length(); i++)
            {
                System.out.println(s.substring(s.length()-i));
            }
        }
    }
}
```

◆ Q25 Shift characters (A→C)

```
class Q25
{
    public static void main(String args[])
    {
        String s = "ABCXYZ";
        String result = "";

        for(int i=0; i<s.length(); i++)
        {
            char ch = s.charAt(i);

            if(ch == 'Y')
                result += 'A';
            else if(ch == 'Z')
                result += 'B';
            else
                result += (char)(ch+2);
        }

        System.out.println(result);
    }
}
```

◆ Q26 First letters word

```
class Q26
{
    public static void main(String args[])
    {
        String s = "Vital Information Resource Under Seize";
        String words[] = s.split(" ");
        String result = "";

        for(int i=0; i<words.length; i++)
        {
```

```
            result += words[i].charAt(0);
        }

        System.out.println(result);
    }
}
```

◆ Q27 Maximum occurring character

```
class Q27
{
    public static void main(String args[])
    {
        String s = "INDIA IS MY COUNTRY";
        int max=0;
        char res=' ';

        for(int i=0; i<s.length(); i++)
        {
            int count=0;

            for(int j=0; j<s.length(); j++)
            {
                if(s.charAt(i)==s.charAt(j))
                    count++;
            }

            if(count>max && s.charAt(i)!=' ')
            {
                max=count;
                res=s.charAt(i);
            }
        }

        System.out.println("Max char: "+res);
    }
}
```

◆ Q28 Remove repeated letters

```
class Q28
{
    public static void main(String args[])
    {
        String s = "india";
        String result = "";

        for(int i=0; i<s.length(); i++)
        {
            char ch = s.charAt(i);

            if(result.indexOf(ch) == -1)
                result += ch;
        }
```

```
        System.out.println(result);
    }
}
```

◆ Q29 Common alphabets

```
class Q29
{
    public static void main(String args[])
    {
        String s1 = "INDIA";
        String s2 = "DELHI";

        for(int i=0; i<s1.length(); i++)
        {
            char ch = s1.charAt(i);

            if(s2.indexOf(ch) != -1)
            {
                System.out.print(ch + " ");
            }
        }
    }
}
```

◆ Q30 Frequency of a word in sentence

```
class Q30
{
    public static void main(String args[])
    {
        String s = "THIS IS A TEST THIS IS ONLY TEST";
        String word = "THIS";

        String arr[] = s.split(" ");
        int count = 0;

        for(int i=0; i<arr.length; i++)
        {
            if(arr[i].equals(word))
                count++;
        }

        System.out.println("Frequency: " + count);
    }
}
```

✓ D. Assertion & Reason

Q1

A = True

R = False

✓ Answer: c

Q2
A = True
R = True (correct explanation)
✅ Answer: a

Q3
A = True
R = True
✅ Answer: a

✅ E. Case-based Questions

Q1 String array
✅ Answer: c

Q2 Declaration
✅ Answer: c (Both)

Q3 Static initialization
✅ Answer: a

Q4 Initialization types
✅ Answer: c (Both)

Q5 Size of array
Given array of 3 elements
✅ Answer: d (3)

🎯 Final Note for Students

- **String is immutable** (cannot change directly)
- Always remember:
  - equals() → compare
  - + → join
  - trim() → remove spaces
  - toUpperCase() → uppercase

****