upGrad
#LifeKoKaroLift

PGC Full Stack Development

06/10/2020

upGrad

Course: Foundation of programming

Lecture on: Strings and Arrays

Instructor: Rajesh Kumar

In Previous Class, We Covered…

1 Loops in Java

2 Different types of loops: While, Do-While and For

3 Break and Continue statements

4 Methods

Today’s Agenda

1 String Expressions

2 Operations in Strings: Concatenation, Comparisons and Case Length

3 Arrays: Initialisation, size and accessing elements

4 Types of Arrays: 1D and 2D

Yesterday’s Homework

Write a program to print the following pattern:

```
1*2*3*4*17*18*19*20
 5*6*7*14*15*16
  8*9*12*13
   10*11
```

Input Format:

No input required

Output Format:

First line containing 1*2*3*4*17*18*19*20

Second line containing 5*6*7*14*15*16

Third line containing 8*9*12*13

Fourth line containing   10*11

Yesterday's Homework Solution

```
package com.company;
import java.util.Scanner;
public class Main {

   public static void main(String args[]) {
     Scanner sc = new Scanner(System.in);
     int assetTrackingNumber = sc.nextInt();
     String assetName = sc.next();
     float assetValue = sc.nextFloat();
     System.out.println(assetTrackingNumber);
     System.out.println(assetName);
     System.out.println(assetValue);
    }
}
```

String Expressions

upGrad

- Strings are a sequence of characters.
- In Java, the String class is used to create and manipulate strings.

Initialising a String

There are two ways to initialise a string:

- String name; //Null value
- String name = “John”;

Note: A string has double quotes (“”), whereas a character has single quotes (‘’).

Null

upGrad

- ‘null’ is a reserved word in Java. It seems like a keyword, but it is actually quite similar to 'true' and 'false'.
- ‘null’ is case-sensitive and literal in Java, and because keywords are case-sensitive in Java, we cannot write it as ‘NULL’ or ‘0’.
- It is used as a default value for the uninitialised variable of reference types such as the object or user-defined class. ‘null’ is not used as a default value for any variable of primitive types such as int and float.
- Typecasting ‘null’ to any reference type is allowed at both compile time and runtime. The program will not throw any error or exception.
- We cannot call a non-static method on a reference variable with the ‘null’ value; it will throw the NullPointerException. However, we can call a static method with reference variables with null values.

String Immutability

upGrad

- In Java, string objects are immutable. ‘Immutable’ simply means unmodifiable or unchangeable.

- Once a string object is created, its data or state cannot be changed. However, a new string object can be created. Observe the code below, here we are not modifying the original string object but creating a new one.

```
public static void main(String args[]) {
  String s1="Java";
  s1.concat(" Programming");
  System.out.println(s1);//Java Programming
}
```

- In the code snippet above, Java is not changed. Instead, a new object is created with JavaProgramming. This is why string objects are considered immutable.

Poll 1 (15 sec.)

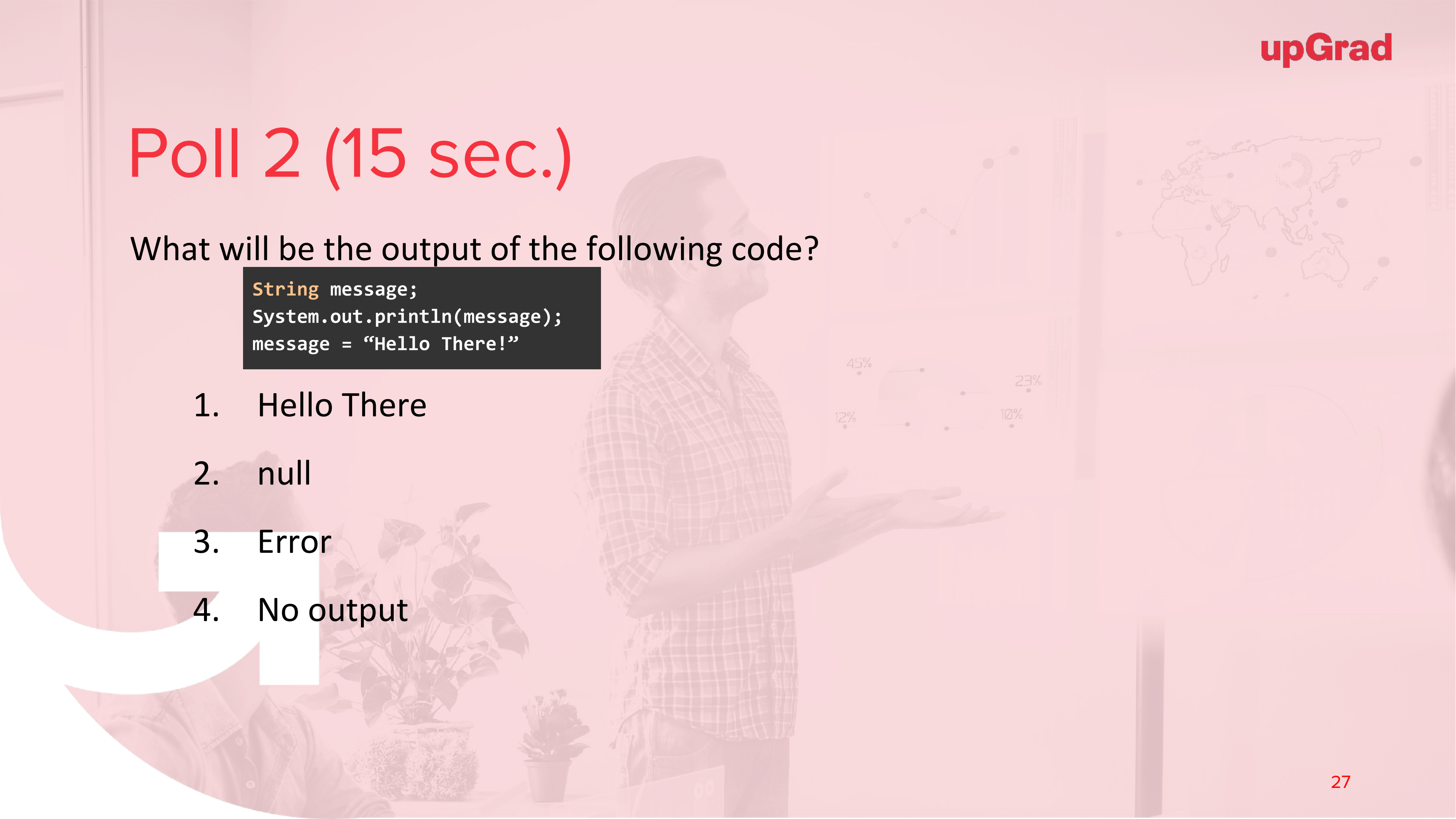
Is a String mutable in Java?

1. Yes
2. No

Poll 1 (Answer)

Is a String mutable in Java?

1. Yes
2. **No**

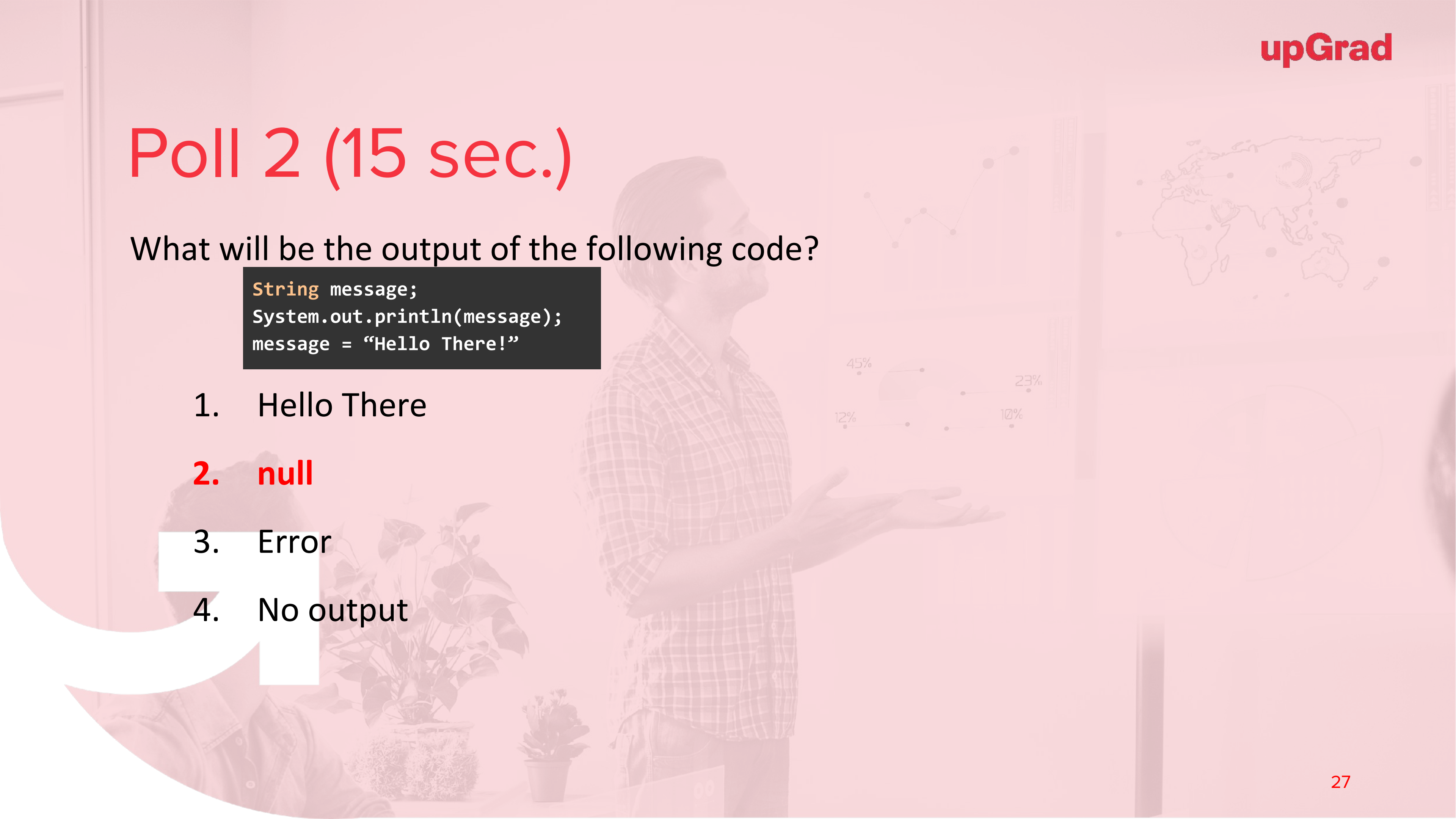


Poll 2 (15 sec.)

What will be the output of the following code?

```
String message;
System.out.println(message);
message = “Hello There!”
```

1. Hello There
2. null
3. Error
4. No output



Poll 2 (15 sec.)

What will be the output of the following code?

```
String message;
System.out.println(message);
message = “Hello There!”
```

1. Hello There
2. **null**
3. Error
4. No output

Hands-On: Exercise 1

upGrad

Declare a string to store a user's name and a character to store the user’s gender.

String Comparison

upGrad

- In Java, a string can be compared on the basis of content and reference.

- String comparison can be used in the following:
  - Authentication (using the equals() method)
  - Sorting (using the compareTo() method)
  - Reference matching (using the == operator)

- There are three ways to compare a string in Java:
  - Using the equals() method
  - Using the = = operator
  - Using the compareTo() method
- equals() method:

```
public static void main(String args[]) {
  String s1="Java";
  String s2="JAVA";
  System.out.println(s1.equals(s2));//false
}
```

String Comparison

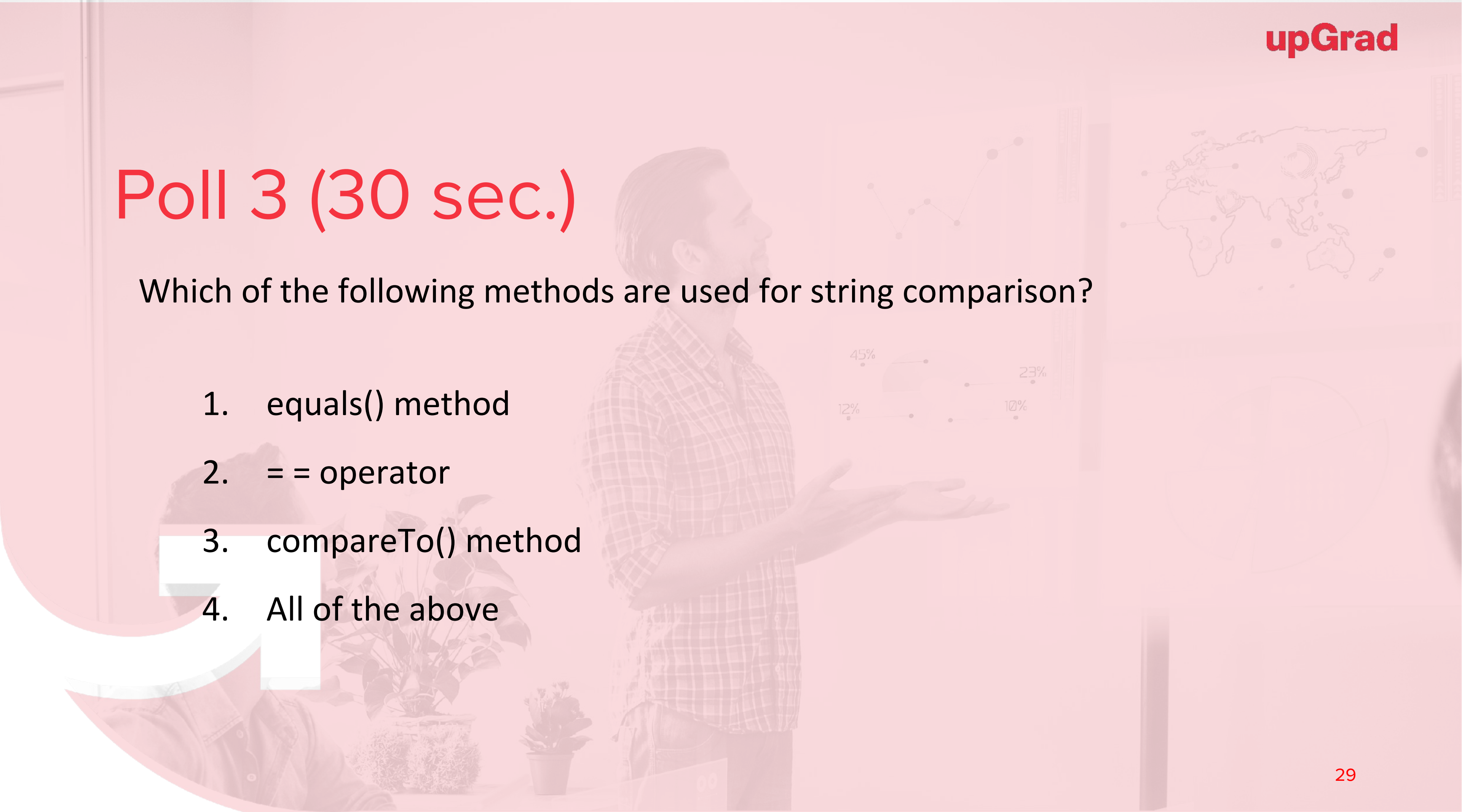
upGrad

- = = operator:

```
public static void main(String args[])
{
  String s1="Java";
  String s2="Java";
  System.out.println(s1==s2);//true
}
```

- compareTo() method:

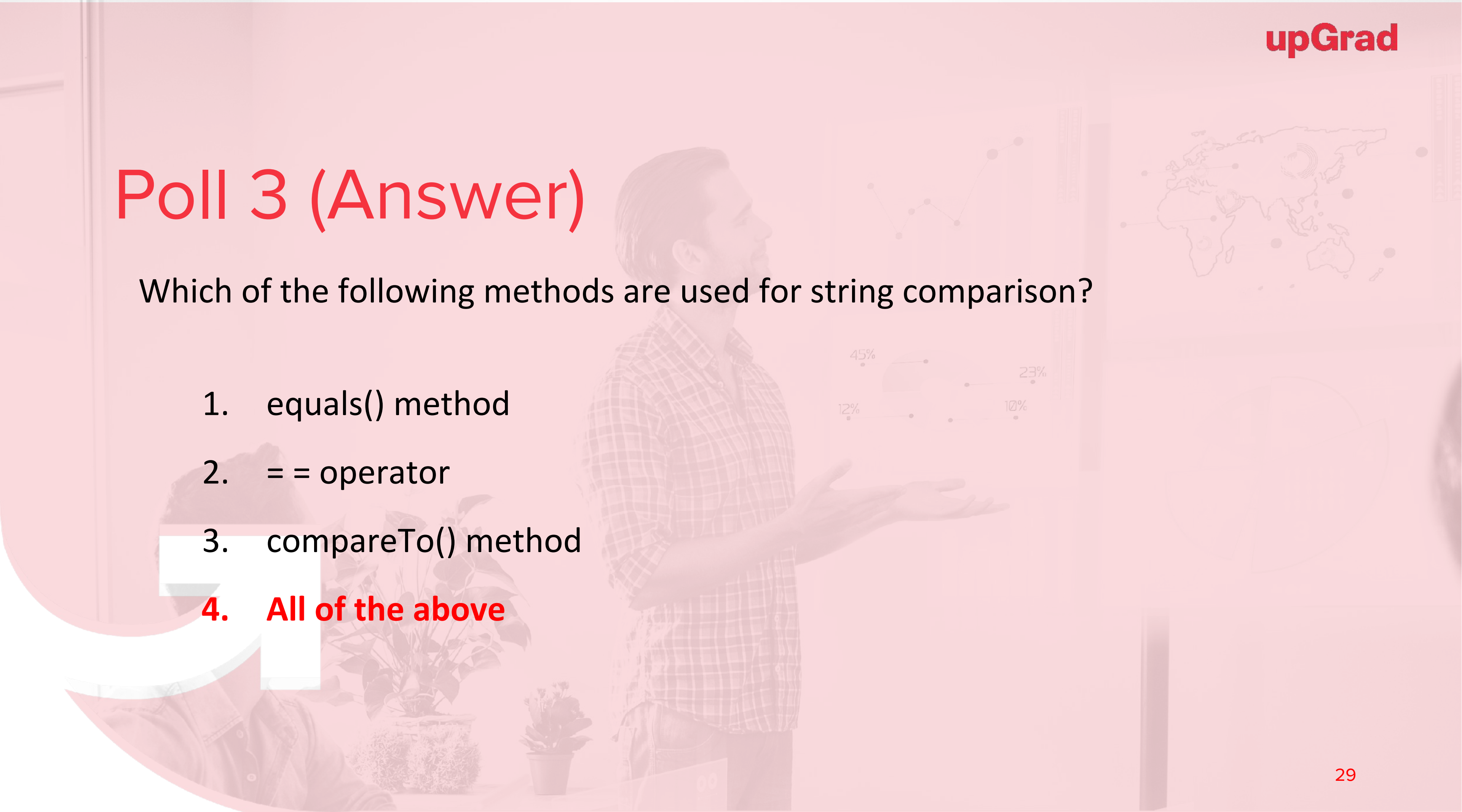
```
public static void main(String args[]) {
  String s1="Java";
  String s2="JAVA";

System.out.println(s1.compareTo(s2));//false
}
```



Poll 3 (30 sec.)

Which of the following methods are used for string comparison?

1. equals() method
2. = = operator
3. compareTo() method
4. All of the above



upGrad

Poll 3 (Answer)

Which of the following methods are used for string comparison?

1. equals() method
2. = = operator
3. compareTo() method
4. **All of the above**

String Case

upGrad

- You can convert the characters of a string into the opposite case using the following methods:

From Uppercase to Lowercase

- The toLowerCase() method converts the characters of a String into lower-case characters.

```
public static void main(String args[]) {
  String s1="JAVA";
  System.out.println(s1.toLowerCase(s1));
}
```

From Lowercase to Uppercase

- The toUpperCase() method converts the characters of a String into upper-case characters.

```
public static void main(String args[]) {
  String s1="java";
  System.out.println(s1.toUpperCase(s1));
}
```

String Length

upGrad

- The length of a string is refers to the total number of characters it contains.
- To calculate the length of a string in Java, you can use the inbuilt length() method of the Java string class.
- The method length() counts the total number of characters in a given String.

```
public static void main(String args[]) {
  String s1 = “Heya”;
  int length = s1.length();
  //Length of the string is printed.
  System.out.println(length);//4
}
```

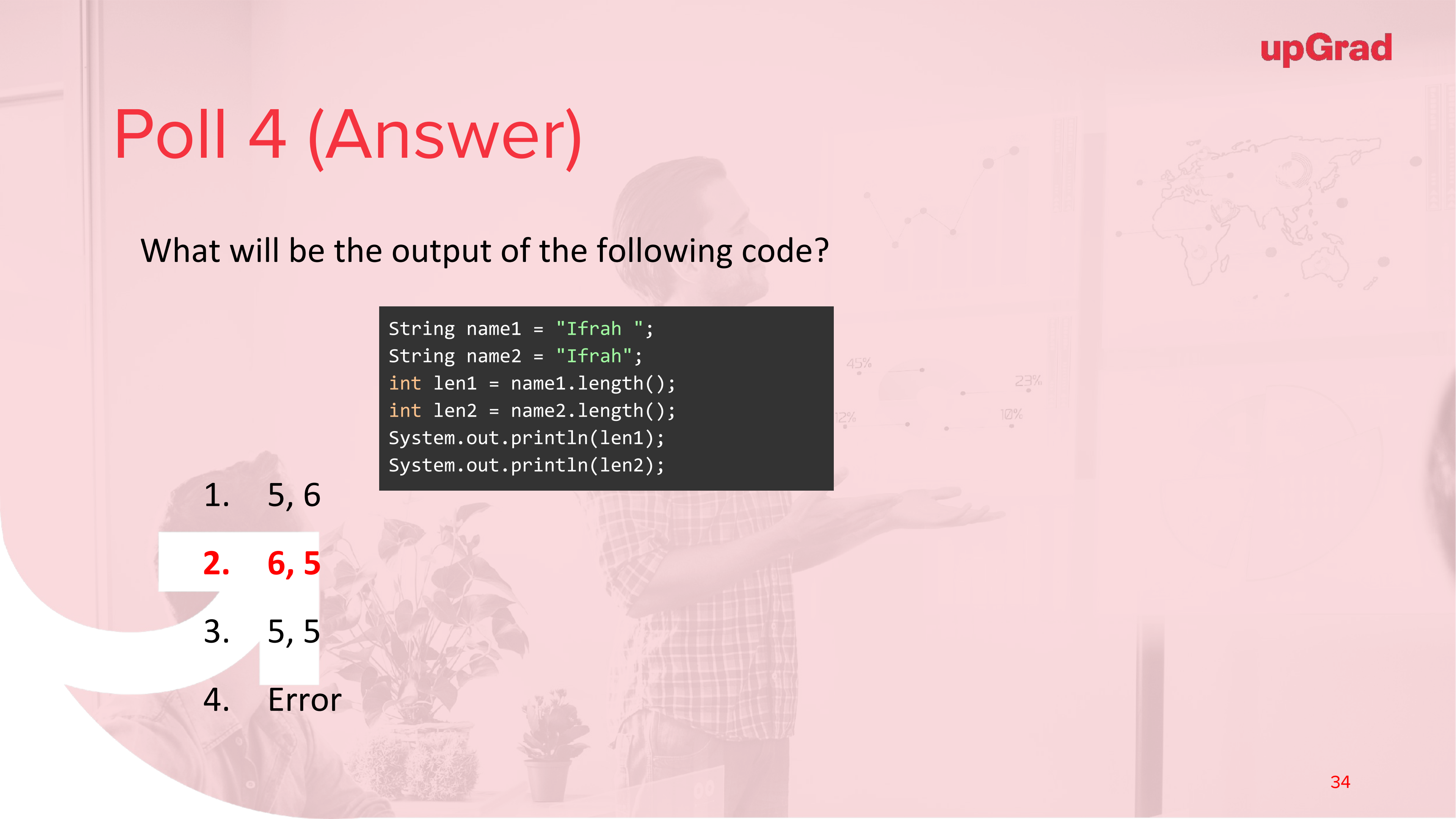
upGrad

Poll 4 (30 sec.)

What will be the output of the following code?

```
String name1 = "Ifrah ";
String name2 = "Ifrah";
int len1 = name1.length();
int len2 = name2.length();
System.out.println(len1);
System.out.println(len2);
```

1. 5, 6
2. 6, 5
3. 5, 5
4. Error



Poll 4 (Answer)

What will be the output of the following code?

```
String name1 = "Ifrah ";
String name2 = "Ifrah";
int len1 = name1.length();
int len2 = name2.length();
System.out.println(len1);
System.out.println(len2);
```

1. 5, 6
2. **6, 5**
3. 5, 5
4. Error

Hands-On: Exercise 2

Write a Java program to perform the following operations:

1. Take an input string in upper case.
2. Convert the case of the string into lower case.
3. Determine the length of the string.

String Concatenation

- String concatenation is the process of forming new string that by a combination of multiple strings.

- There are two ways to perform string concatenation in Java:
  - Using the + (string concatenation) operator
  - Using the concat() method

- + (string concatenation) operator:

```
public static void main(String args[]) {
  String s1="Java";
  s1.concat(" Programming");
  System.out.println(s1);//Java Programming
}
```

- concat() method:

```
public static void main(String args[]) {
  String s1="Java" + " Programming";
  System.out.println(s1);//Java
Programming
}
```

Java String format()

upGrad

- In Java, we use the String.format() method to format strings.
- The method returns the formatted string by the given locale, format and argument.
- If you do not specify the locale in the String.format() method, it uses the default locale by calling the Locale.getDefault() method.

Signature

- There are two types of the method:
  - public static String format(String format, Object... args)
  - public static String format(Locale locale, String format, Object... args)

Parameters

- locale: The locale to be applied on the format() method
- format: The format of the string
- args: The arguments for the format string. It may be zero or more.

Returns

- It returns the formatted strings.

Java String format() - Example

upGrad

```java
public static void main(String args[]) {
  String s1 = String.format("%d", 101);
  String s2 = String.format("%s", "Java");
  String s3 = String.format("%f", 101.00);
  String s4 = String.format("%x", 101);
  String s5 = String.format("%c", 'c');
  System.out.println(s1);
  System.out.println(s2);
  System.out.println(s3);
  System.out.println(s4);
  System.out.println(s5);
}
```

Poll 5 (30 sec.)

What will be the output of the following code?

```
String a = "Welcome";
String b = "Home";
System.out.println(a.concat(b));
```

1. WelcomeHome
2. Welcome Home
3. Home Welcome
4. HomeWelcome

Poll 5 (Answer)

What will be the output of the following code?

```
String a = "Welcome";
String b = "Home";
System.out.println(a.concat(b));
```

1. **WelcomeHome**
2. Welcome Home
3. Home Welcome
4. HomeWelcome

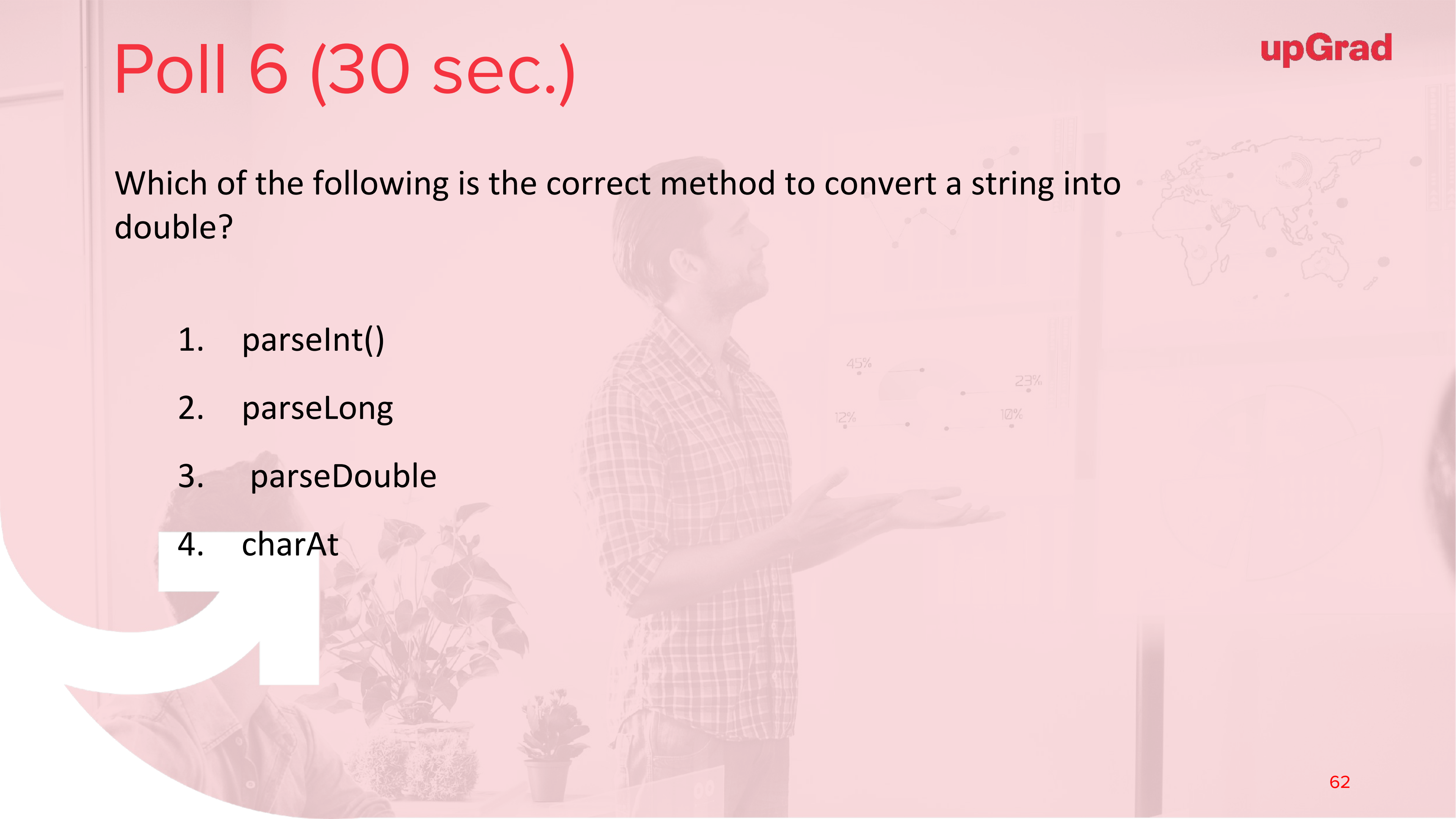
Converting Primitive Data Type into a String

upGrad

- In Java, the string.valueOf() method converts different types of values into strings.
- Using the string.valueOf() method, you can convert int into string, long into string, boolean into string, character into string, float into string, double into string, object into string and char array into string.
- The following example demonstrates the conversion of an integer to string.

```
public static void main(String args[]){
int number=10;
String s1=String.valueOf(value);
System.out.println(s1);
}
```

Poll 6 (30 sec.)



Which of the following is the correct method to convert a string into double?

1. parseInt()
2. parseLong
3. parseDouble
4. charAt

Poll 6 (30 sec.)

Which of the following is the correct method to convert a string into double?

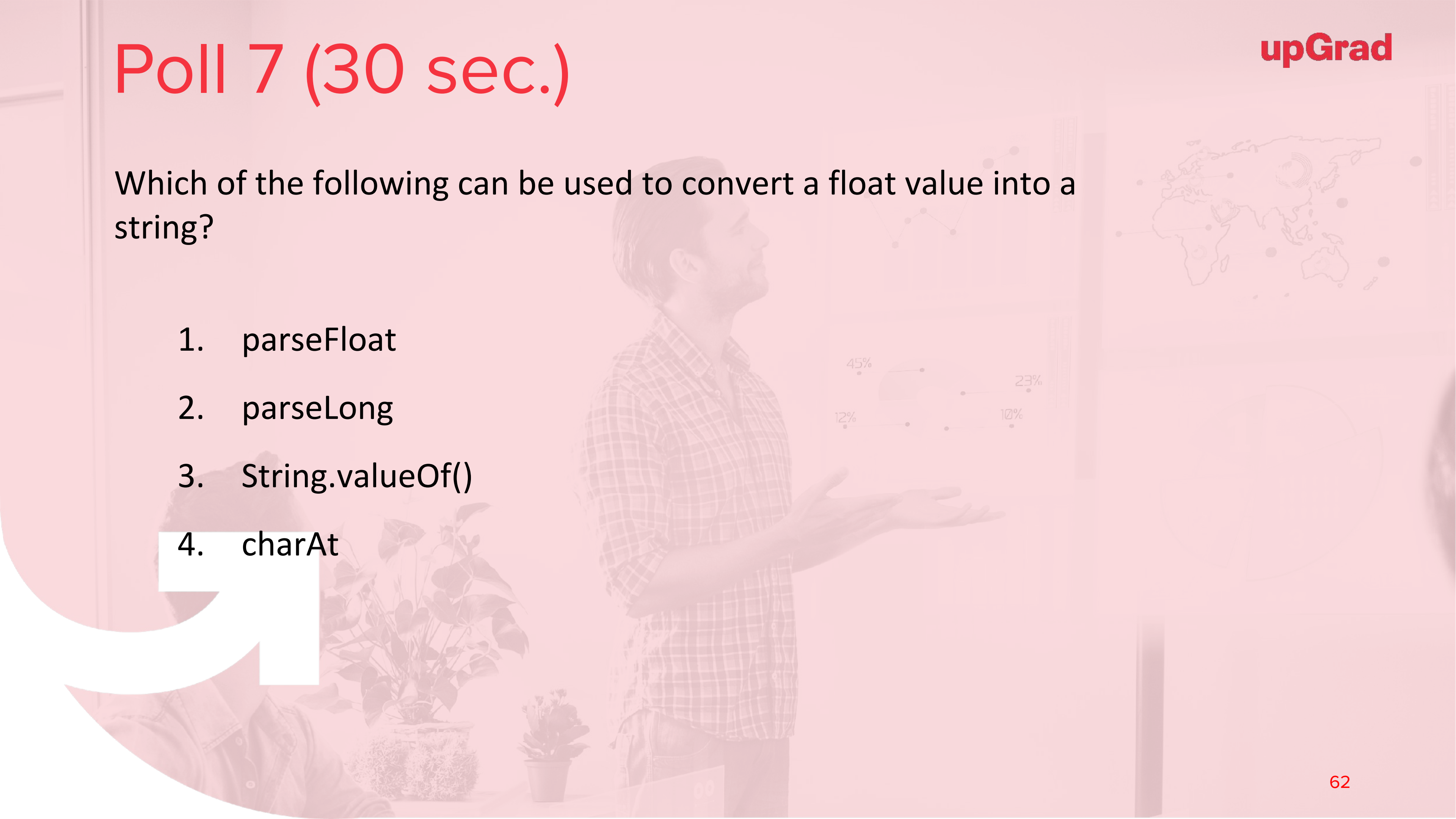
1. parseInt()
2. parseLong
3. **parseDouble**
4. charAt

Poll 7 (30 sec.)

upGrad

Which of the following can be used to convert a float value into a string?

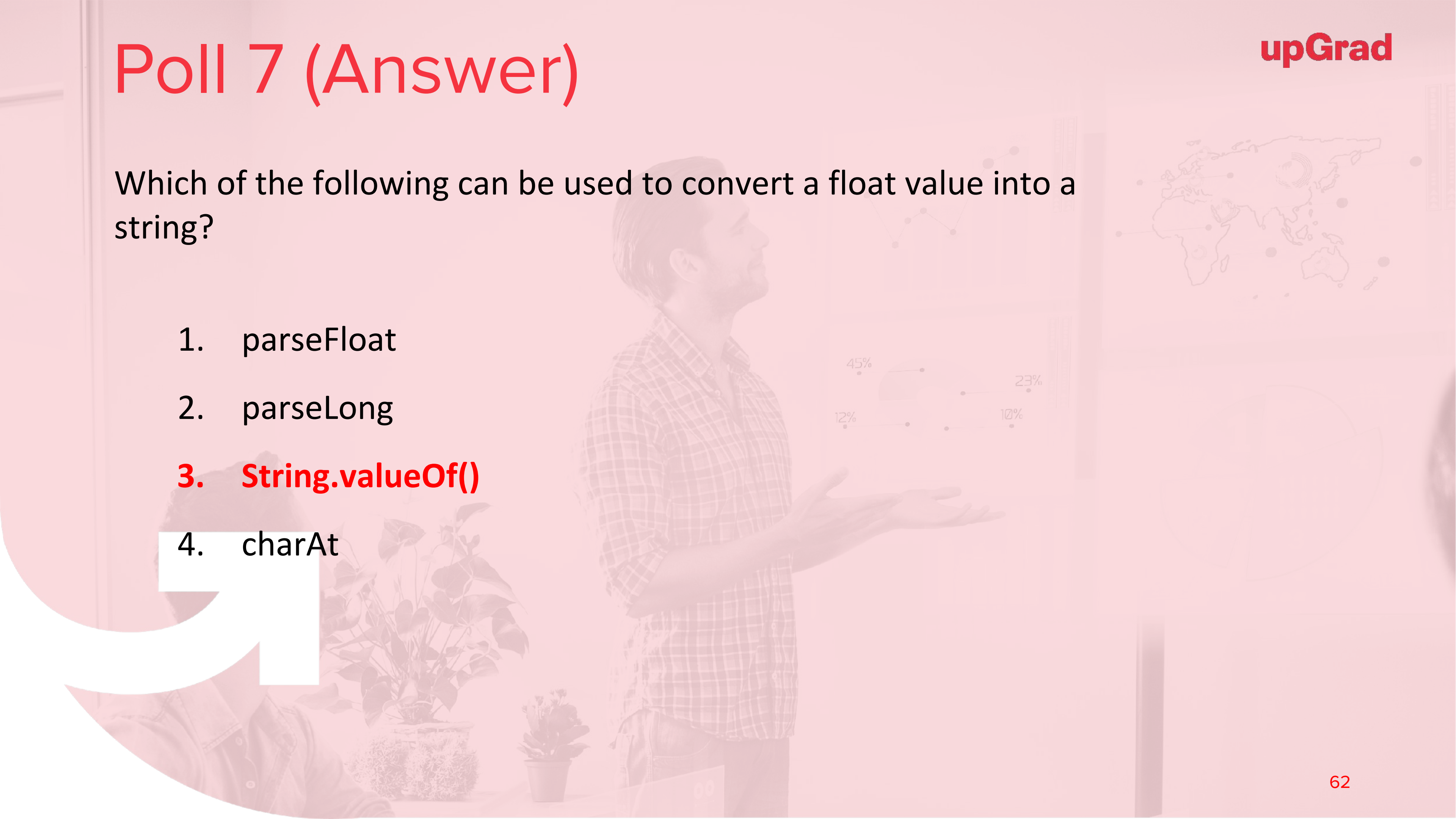
1. parseFloat
2. parseLong
3. String.valueOf()
4. charAt



Poll 7 (Answer)

Which of the following can be used to convert a float value into a string?

1. parseFloat
2. parseLong
3. **String.valueOf()**
4. charAt



StringBuffer and StringBuilder

upGrad

- In Java, there are three classes to represent a sequence of characters: String, StringBuffer, and StringBuilder.
- The String class is an immutable class, whereas StringBuffer and StringBuilder classes are mutable.
- **StringBuffer:** The StringBuffer class is used to create mutable (modifiable) strings. In Java, the StringBuffer class is same as the String class, except the fact that the former is mutable, i.e., it can be changed.

```
public static void main(String args[]){
StringBuffer s1=new StringBuffer("Java ");
s1.append(" Programming");//now original string is
changed
System.out.println(s1);//prints Java Programming
}
```

- **StringBuilder:** Similar to the StringBuffer case, the StringBuilder class is also used to create mutable strings

```
public static void main(String args[]){
StringBuilder s1=new StringBuffer("Java ");
s1.append(" Programming");//now original string is
changed
System.out.println(s1);//prints Java Programming
}
```

StringBuffer vs StringBuilder

upGrad

No.	StringBuffer	StringBuilder
1)	StringBuffer is *synchronised,* i.e., thread-safe. This means that two threads cannot call the methods of the StringBuffer class simultaneously.	StringBuilder is *non-synchronised,* i.e., it is not thread-safe. This means that two threads can call the methods of the StringBuilder class simultaneously.
2)	StringBuffer is *less efficient* than StringBuilder.	StringBuilder is *more efficient* than StringBuffer.

Hands-On: Exercise 3

1. Write a program to round off the price of a product to the nearest decimal.
2. Create a string using the StringBuffer class first and then the StringBuilder class.

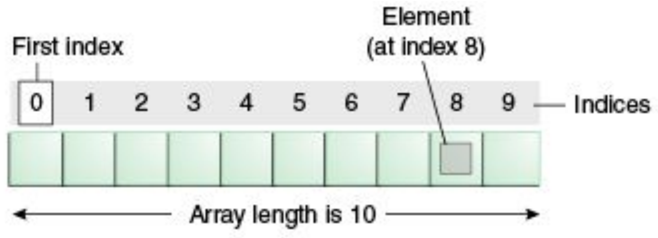
upGrad

Arrays

Arrays

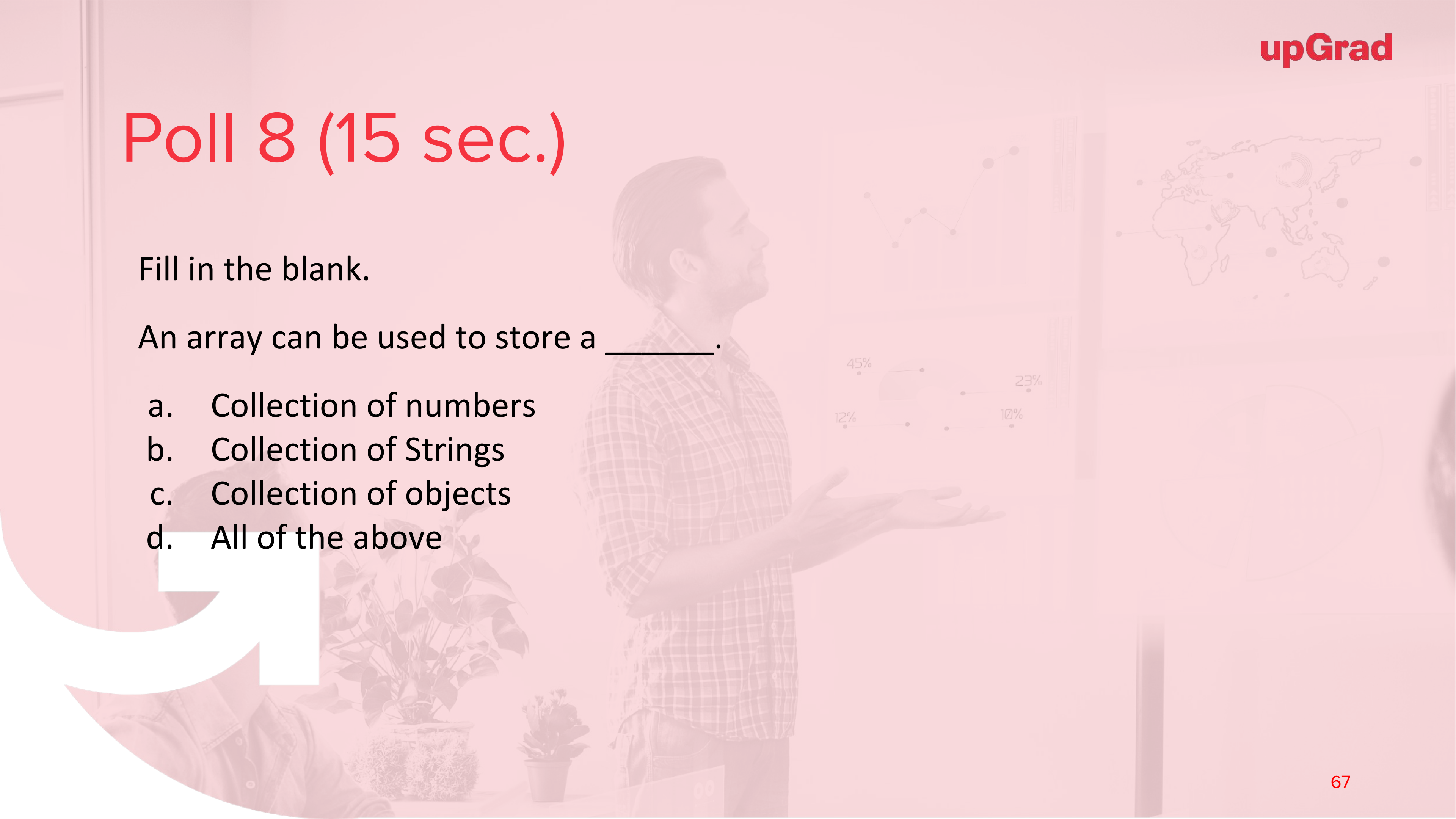
upGrad

- An array is a container object that holds a predefined and fixed number of values of a single type.
- The length of an array is fixed, after it is created.
- Each entity in an array is called an element, and each element is accessed by its numerical index, as shown in the image below.



An array of 10 elements.

- An array can be one dimensional or multidimensional.

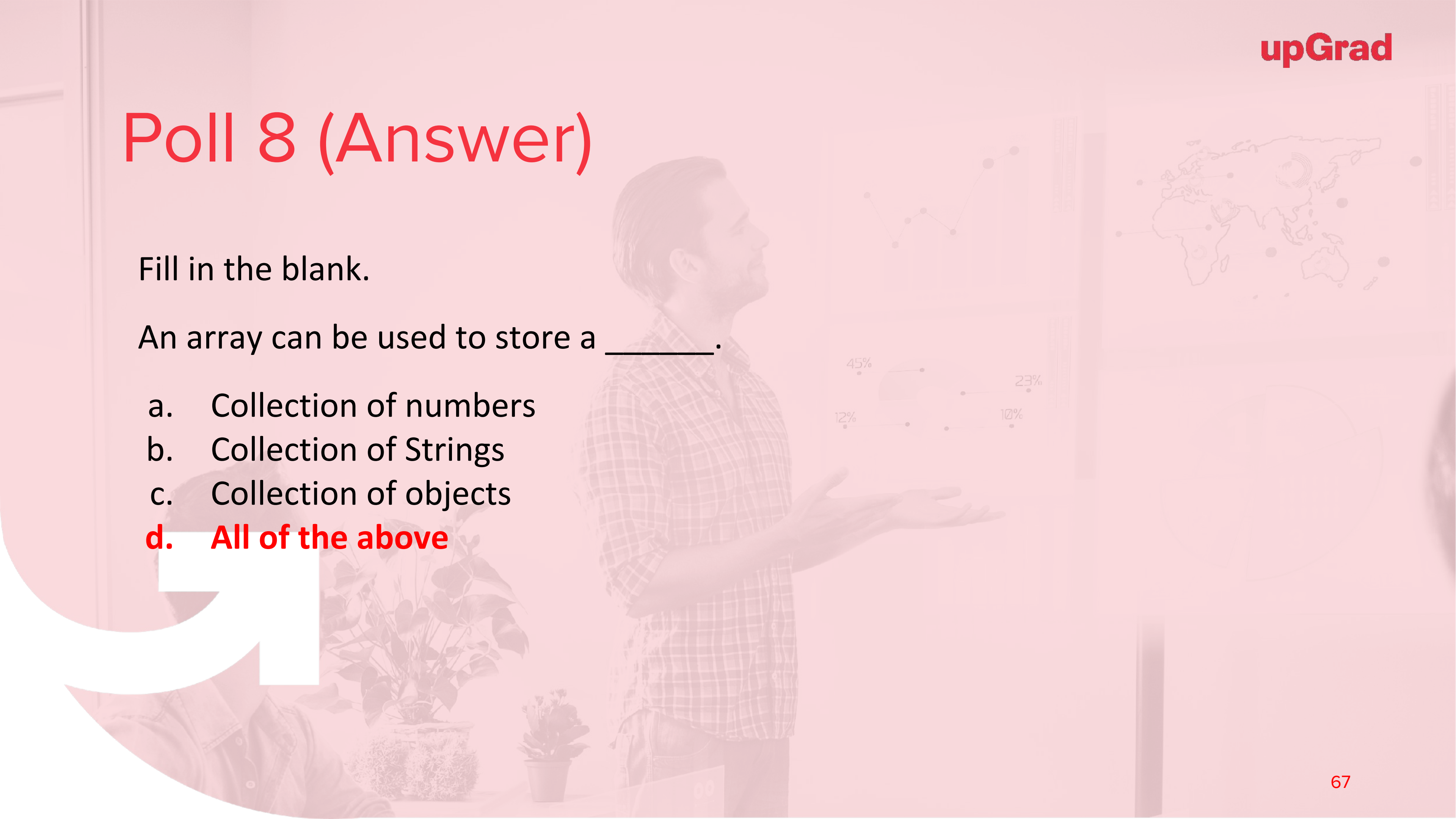


Poll 8 (15 sec.)

Fill in the blank.

An array can be used to store a _______.

a. Collection of numbers
b. Collection of Strings
c. Collection of objects
d. All of the above



Poll 8 (Answer)

Fill in the blank.

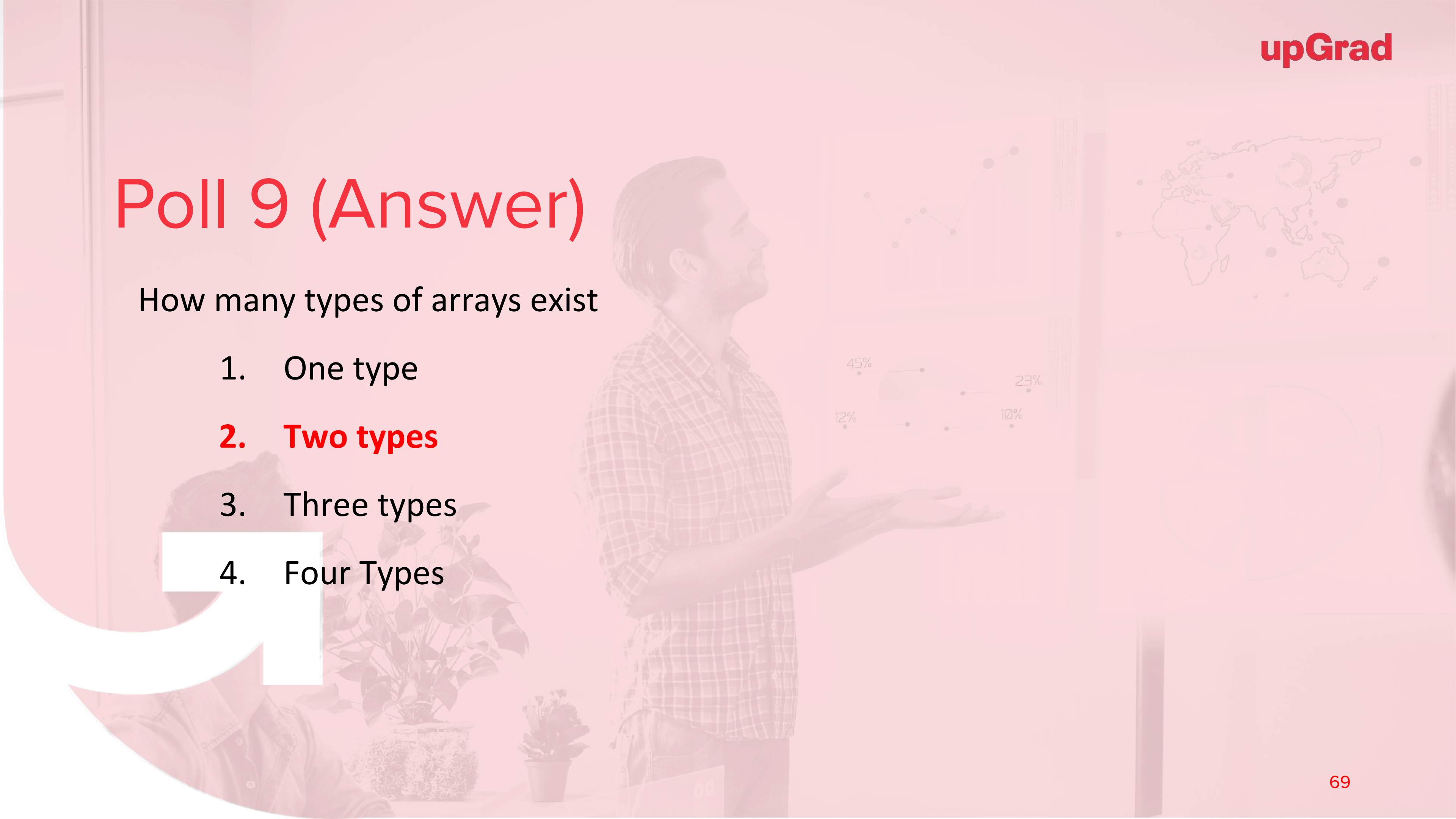
An array can be used to store a ______.

a. Collection of numbers
b. Collection of Strings
c. Collection of objects
d. All of the above

Poll 9 (30 sec.)

How many types of arrays exist?

1. One type
2. Two types
3. Three types
4. Four Types



Poll 9 (Answer)

How many types of arrays exist

1. One type
2. **Two types**
3. Three types
4. Four Types

1D Arrays

upGrad

- Creating an Array:

```
// declares an array of integers
int[] anArray;
```

- Initialising an Array:

```
anArray[0] = 100; // initialise first element
anArray[1] = 200; // initialise second element
anArray[2] = 300; // and so forth
```

- Accessing elements:

```
System.out.println("Element 1 at index 0: " + anArray[0]);
System.out.println("Element 2 at index 1: " + anArray[1]);
System.out.println("Element 3 at index 2: " + anArray[2]);
```

ForEach Loop

upGrad

- The foreach loop got introduced in JDK 1.5.
- It is also known as enhanced for loop.
- The foreach loop is used to traverse an array sequentially without using an index.

```java
public class TestArray {
   public static void main(String[] args) {
      double[] myList = {1.9, 2.9, 3.4, 3.5};
      // Print all the array elements
      for (double element: myList) {
         System.out.println(element);
      }
   }
}
```

upGrad

Poll 10 (30 sec.)

Which of the following is the correct syntax for the forEach loop?

1. `for (arrayType element: arrayName)`
2. `for (arrayName element: arrayType)`
3. `forEach (arrayType element: arrayName)`
3. `forEach (arrayName element: arrayType)`

upGrad

Poll 10 (Answer)

Which of the following is the correct syntax for the forEach loop?

1. `for (arrayType element: arrayName)`
2. `for (arrayName element: arrayType)`
3. `forEach (arrayType element: arrayName)`
3. `forEach (arrayName element: arrayType)`

Hands-On: Exercise 4

upGrad

Suppose a few students have scored certain marks in Mathematics, and their marks are stored in an array as follows:

```
mathMarks = [20,40,45,43,29,38,33];
```

You need to determine the highest marks obtained by a student.

2D Arrays

upGrad

- A two-dimensional array or a 2D array is an array of arrays.
- A 2D array is organised as matrices that can be represented in the form of a collection of rows and columns.

Declaring the Array:

```
int arr[rows][columns];
or
int arr[2][2] = {0,1,2,3};
```

Accessing Data in the Array:

```
int x = arr[i][j];
```

Hands-On: Exercise 5

Write a Java program to store the following information related to books, which are sold in an online store, in a 2D array:

1. Title
2. Author Name
3. Publishing House
4. Genre
5. Price

Today’s Homework

Write a program to count the the number of times a character appears in an input string.

Input Format:
The first line will contain the input sentence.
The second line will contain the character to be searched.

Output Format:
One line containing the number of times the the number of times a character appears in an input string

Sample Input 1:
We are learning Java
a
Sample Output 1:
4

Explanation:
The character ‘a’ appears four times.

upGrad
#LifeKoKaroLift

Thank You!