upGrad

DATA STRUCTURES & ALGORITHMS

upGrad

Lecture On: Introduction to Java

Course Roadmap

- **Introduction to Java**
- Operators and Conditionals
- Loops and Methods
- Strings and Arrays

upGrad

Today’s Agenda

Why do we need a programming language?

In our day-to-day lives, we use communication to perform different tasks. For example, at a restaurant, when you want to place an order, you call the waiter and give them instructions. Similarly, to prepare your favourite dish, you follow certain steps.

Computers also work in a similar manner, and you need to provide them with certain instructions in order to use them to solve some important tasks.

A computer understands everything in its own language, which is called the **machine language**, in which all the data is stored in the form of binary digits, known as bits in short.

Since machine language is difficult for humans to understand, there are **high-level** programming languages that are understandable by humans and, later, converted to machine language using the process of compilation.

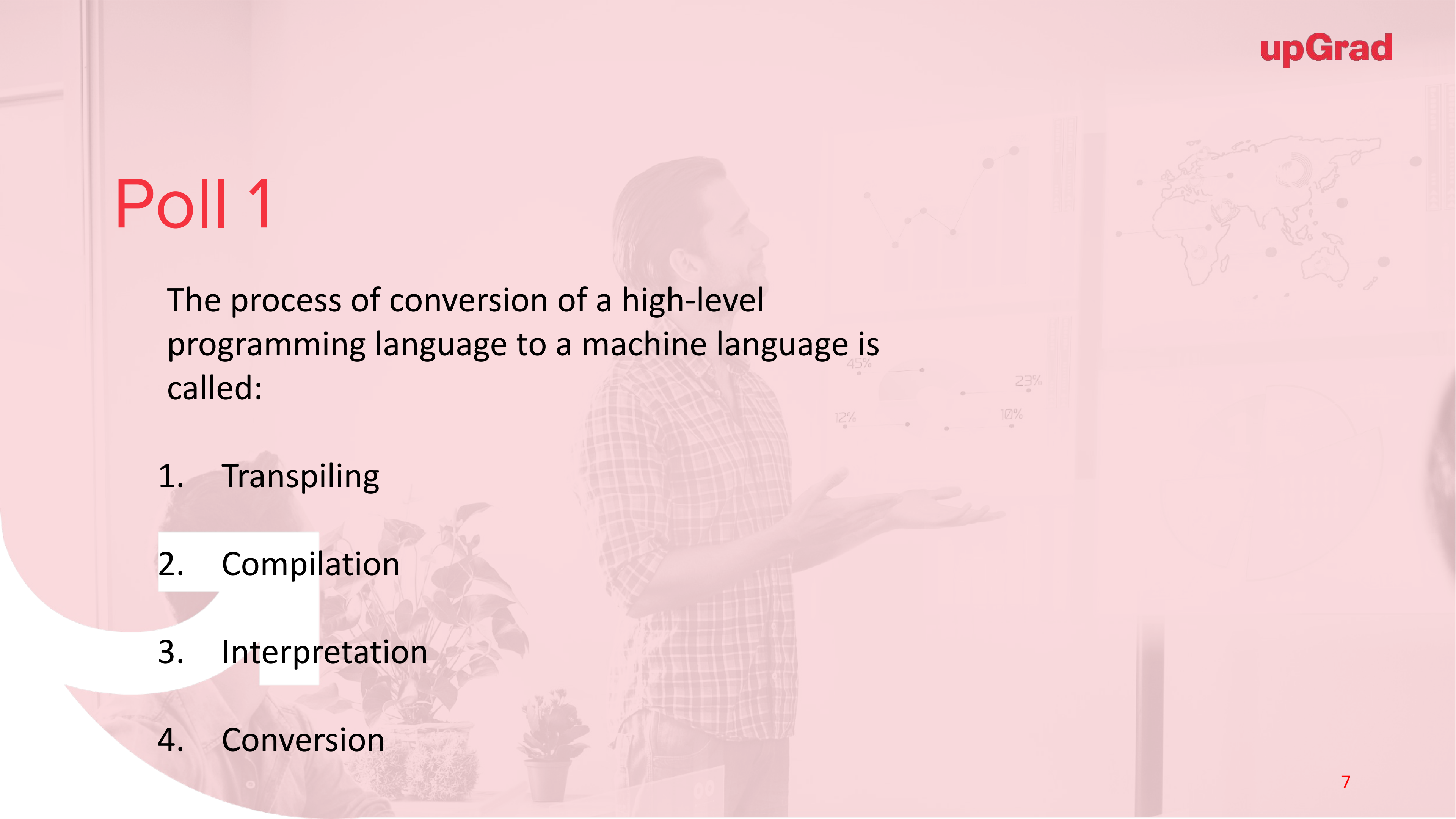
Introduction to Java

There exist many different high-level programming languages, such as:

a. Python
b. Visual Basic
c. Delphi, Perl
d. PHP, ECMAScript
e. Ruby
f. C#
g. Java

What is Java?

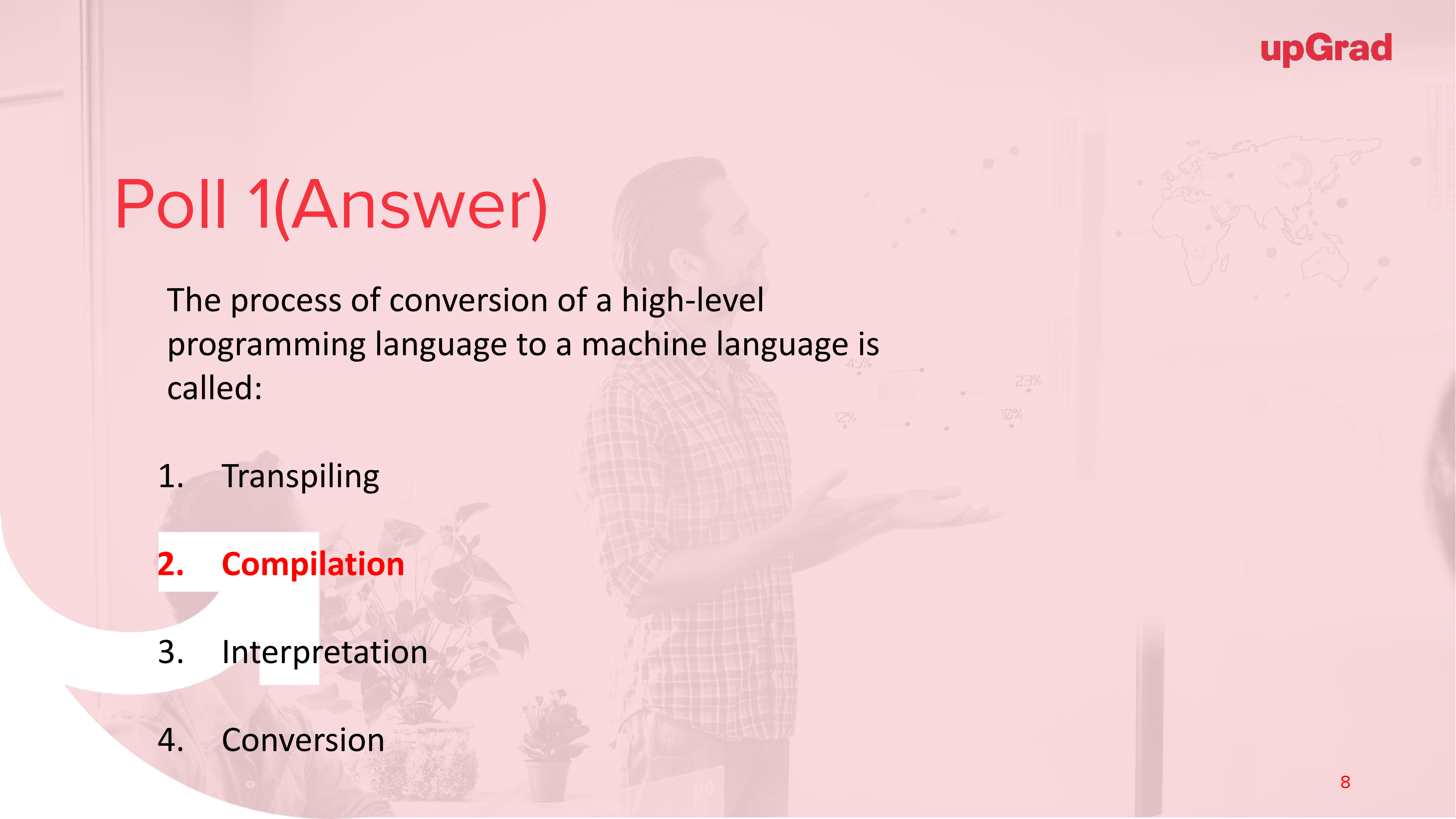
- Java is a general-purpose programming language, which was invented by James Gosling at Sun Microsystems.
- It is intended to let application developers Write Once, Run Anywhere (WORA).
- All the source code is first written in plain text files with the .java extension.



Poll 1

The process of conversion of a high-level programming language to a machine language is called:

1. Transpiling
2. Compilation
3. Interpretation
4. Conversion



Poll 1(Answer)

The process of conversion of a high-level programming language to a machine language is called:

1. Transpiling
2. **Compilation**
3. Interpretation
4. Conversion

Java Development Tools

To develop Java applications, we use the Java Development Kit (**JDK**). It provides the environment to develop and execute Java programs. It consists of the following two components:

- Development Tools – Provide an environment to develop Java programs
- JRE – Execute your Java program

JRE: Java Runtime Environment is an installation package, which provides the environment to run Java programs on your machine. You cannot develop a program in JRE. Hence, it is used only by someone who wants to run a Java programs, i.e., the end users of your program or application. It consists of:

- JVM
- JIT
- Additional classes and packages.

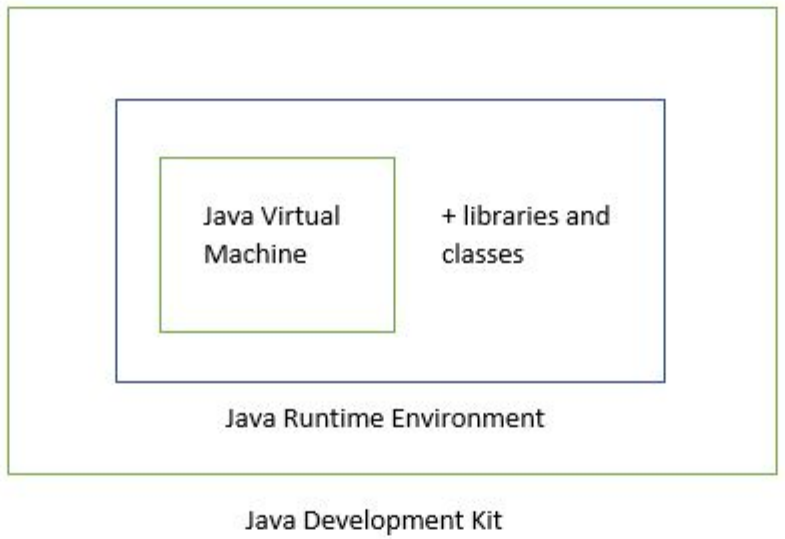
JIT: The Just-In-Time (JIT) compiler is a an essential part of the JRE, which is responsible for performance optimisation of Java-based applications at run time. The compiler is one of the key aspects in deciding the performance of an application for both parties, i.e., the end user and the application developer.

JVM

upGrad

- The **Java Virtual Machine (JVM)** is the virtual machine that runs the Java bytecodes. The JVM does not understand the Java source code, and this is why you need to compile the *.java files to obtain the *.class, which contains the bytecodes, which are understood by the JVM.
- JVM interprets the byte code into machine code depending upon the underlying operating system and hardware combination.
- It is responsible for all the operations, such as garbage collection, array bound checking.
- JVM is platform-dependent.
- It is also the component that allows Java to be a ‘portable language’ (*write once, run anywhere*).

JDK vs JRE vs JVM



Poll 2

Is JVM platform-independent?

1. Yes

2. No

Poll 2(Answer)

Is JVM platform-independent?

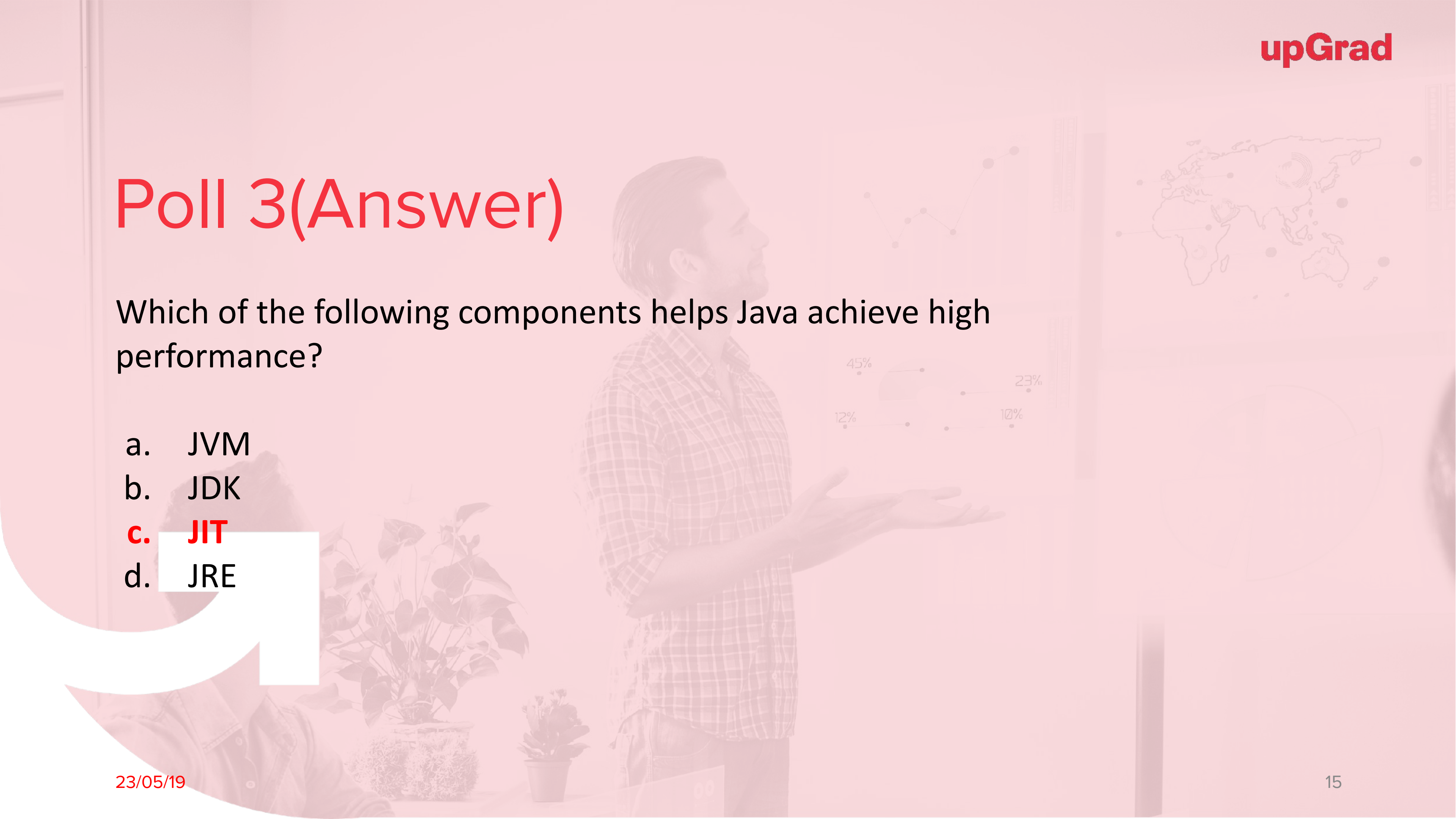
1. Yes

2. **No**

Poll 3

Which of the following components helps Java achieve high performance?

a. JVM
b. JDK
c. JIT
d. JRE



upGrad

Poll 3(Answer)

Which of the following components helps Java achieve high performance?

a. JVM
b. JDK
c. JIT
d. JRE

main() Method

upGrad

Writing a Java Program

Every Java program needs an entry point to execute. The ‘main’ method acts as an entry point in an application and subsequently invokes all other methods present in the program. It is similar to the main method in C/C++.

Signature of main Method

public static void main(String[] args)

public: This is the access modifier of the method, which defines the permission to access the method. The main method has to be public for the program to access it.

static: A method is static when it does not need an object to be called. When the Java program starts, the first component that is encountered is the main method, which does not have any object created. Hence, the method must be static to be called.

main() Method

upGrad

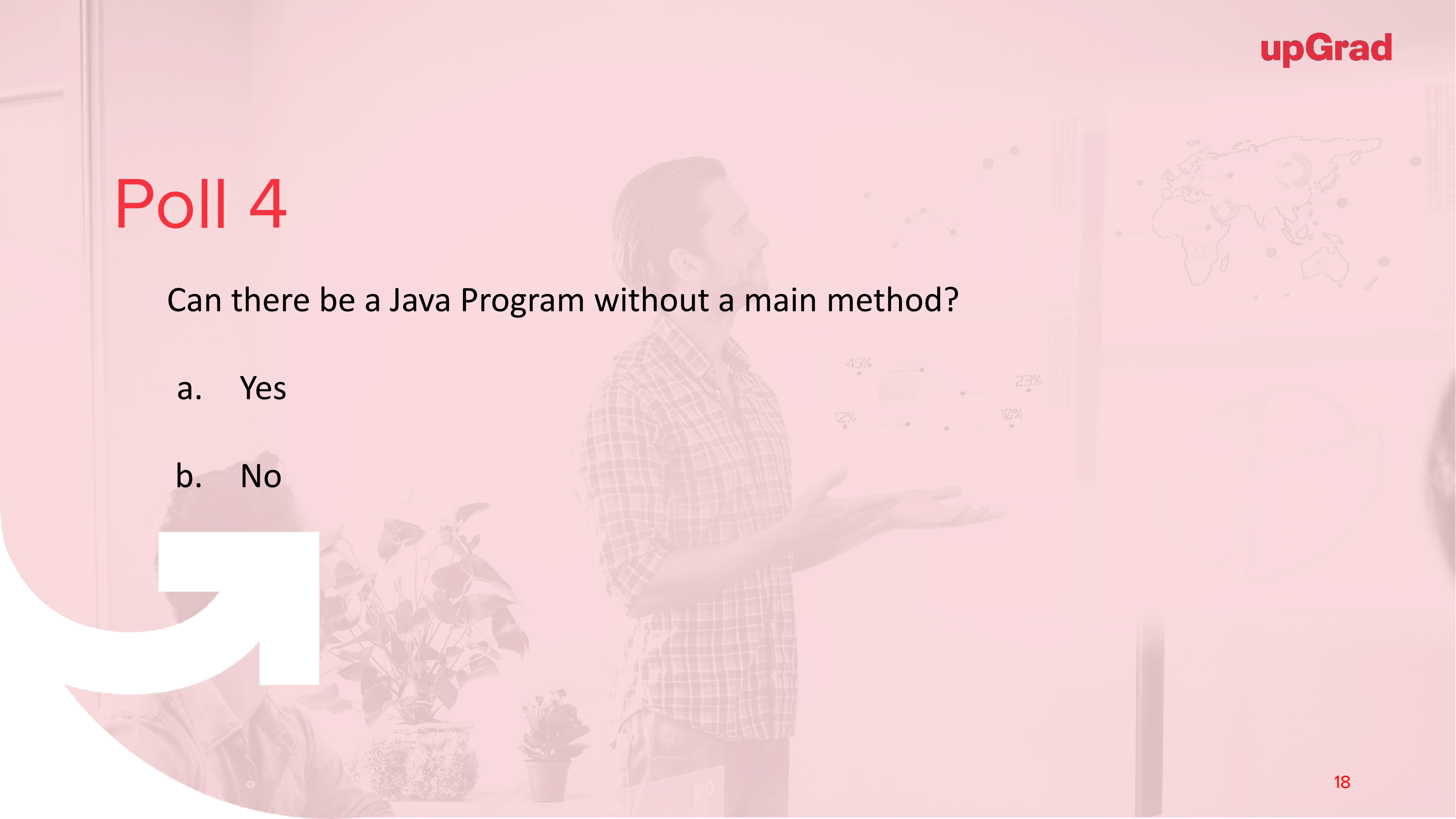
Writing a Java Program

void: A method is void when it does not return anything. Every Java method must have a return type. A Java program ends when the main method is executed and, thus, the method does not return anything.

main: It is the name of the method and should be kept as ‘main’ only'; otherwise, the program will not run.

String[] args: The main method accepts a single argument: An array of elements of type String. Each string in an array is called a **command-line argument**. You can also write this as **String...args.**

Note: You will learn about methods, return type and command-line arguments later in the course.

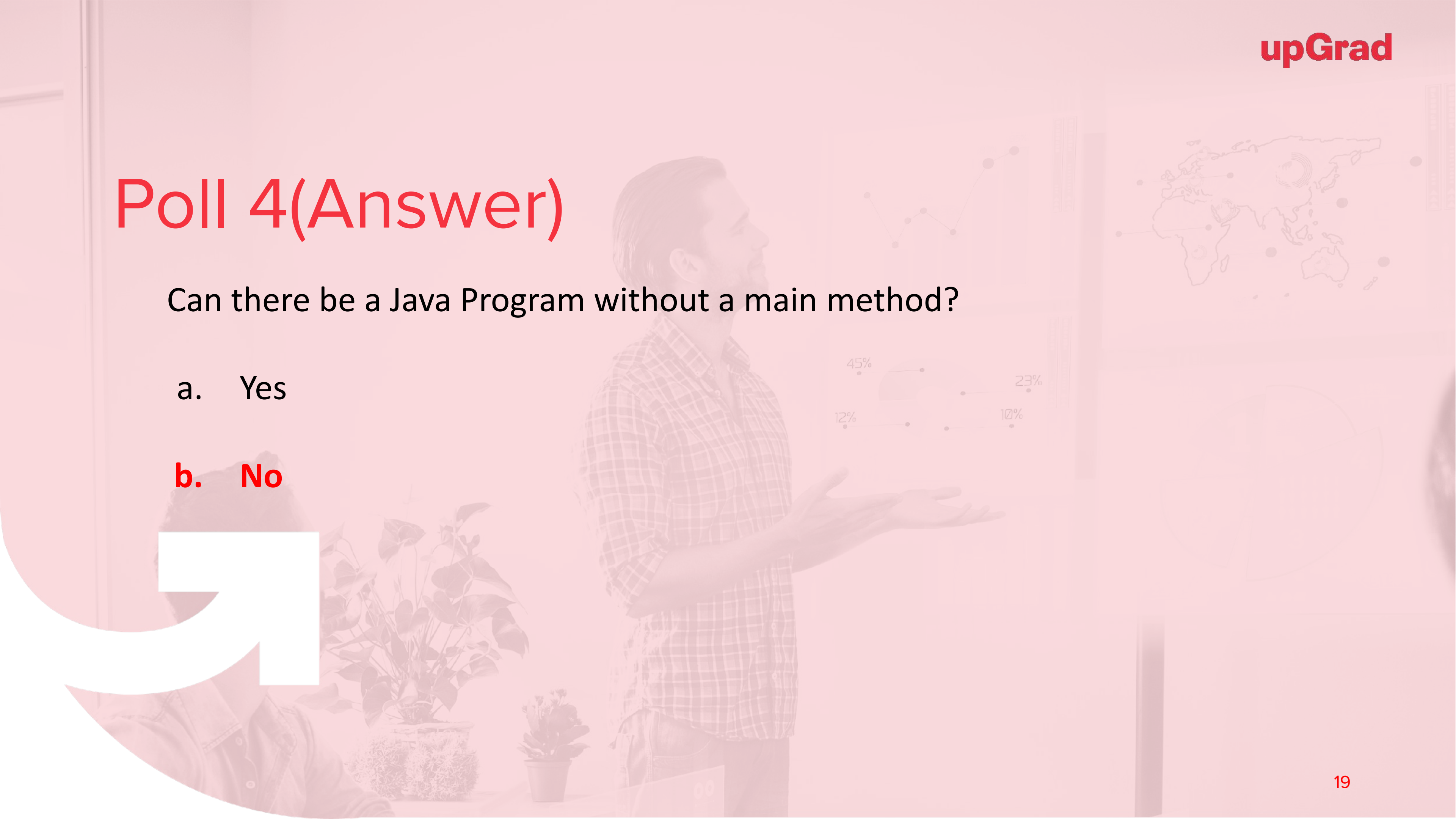


Poll 4

Can there be a Java Program without a main method?

a. Yes

b. No



Poll 4(Answer)

Can there be a Java Program without a main method?

a. Yes

b. No

Variables in Java

upGrad

While developing web applications, you need to store and manipulate a lot of information. For example, Facebook lets you create your profile and stores all the information. When you sign in the next time, it remembers your profile data and fetches your profile for you.

The basic storage unit in a computer program is called a ***variable***. Variables are storage units in your program and they facilitate a major chunk of the computations that you perform.

In simpler terms, variables are containers in the computer memory to store the information that you want. In Java, variables have a type, name and the value they store.

However, to be able to access specific information in the memory, there should be a method to identify these containers uniquely. For this purpose, we use a type of ***identifier*** known as variable names.

Note: Identifiers are a way to refer to variables, classes, etc.

Naming Convention for Variables

upGrad

The variable names should adhere to the following guidelines:

1. They should be closely related to the values they hold.
2. They can include any alphabetic character or digit and an underscore.
3. The cannot have white space.
4. They cannot begin with a number.
5. They cannot be the same as a ***keyword***. A keyword is a reserved word. which is used for a specific purpose and cannot be used elsewhere.

Note:

1. Java is case-sensitive and so, numOne is not the same as NumOne.
2. There is no limit to the length of a variable name in Java.

Variable Declaration

upGrad

- The Java programming language is statically typed, which means all the variables must first be declared before they can be used.
- A variable declaration needs the name of the variable and its data type.

For example:

```
char grade;
```

The code above tells the Java program that a field named grade of type char exists. So, you need to tell the following two information about a variable to the computer:

1. The kind of information that you'd like to store in the variable, so that the computer can allocate memory volume accordingly.
2. The name of the variable, so you can tell the computer to store information in that container or retrieve information from it.

The process of doing so is known as **Variable Declaration**.

Syntax for Variable Declaration:

```
variableType variableName;
```

Variable Initialisation

Once a variable is declared, you need to assign a value to it. The process of assigning a value to a variable is known as **Variable Initialisation**.

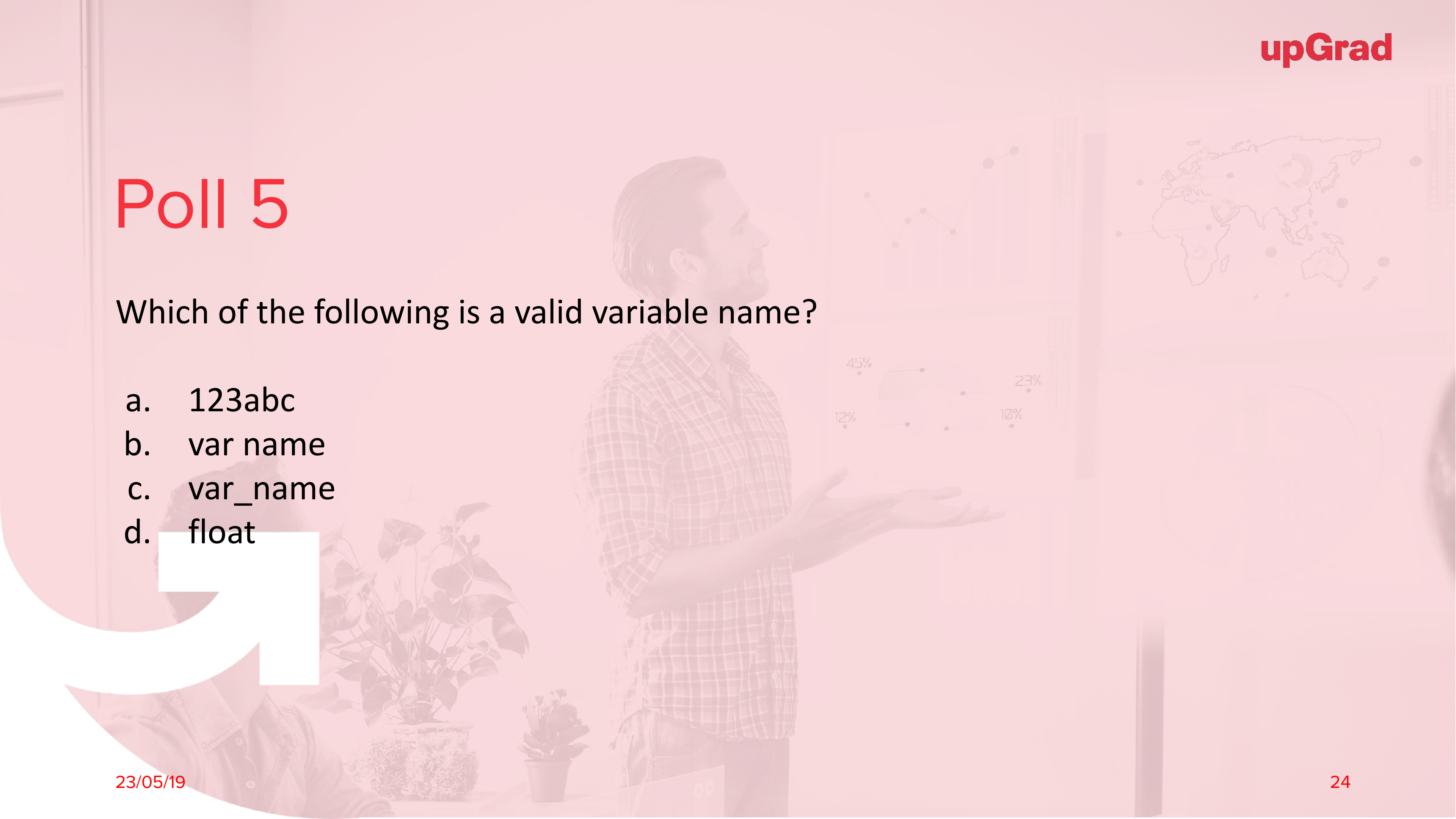
A variable can be initialised as follows:

a. Declared first and initialised later

```
char grade;
grade = 'A';
```

b. Declared and initialised together

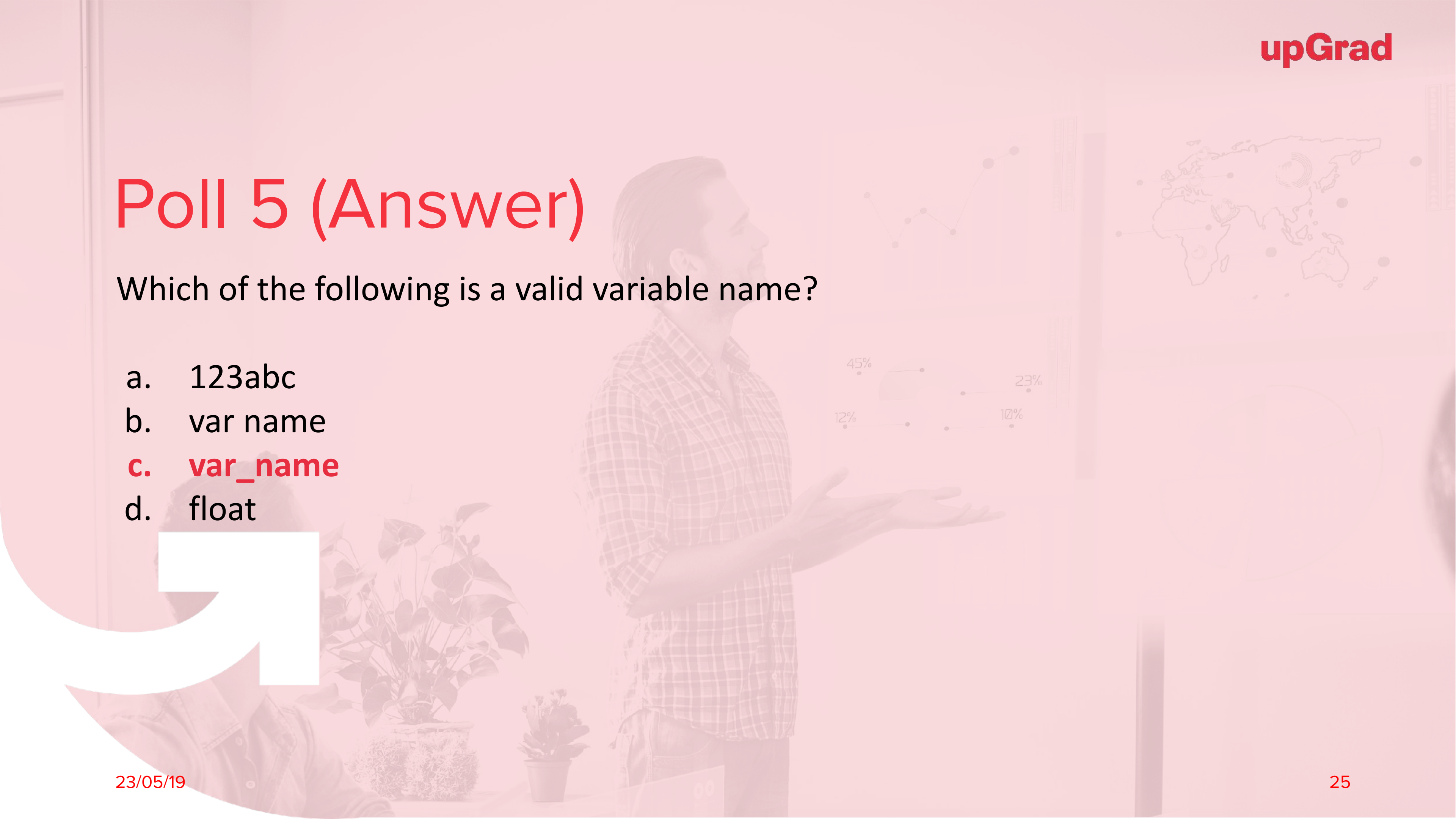
```
char grade = 'A';
```



Poll 5

Which of the following is a valid variable name?

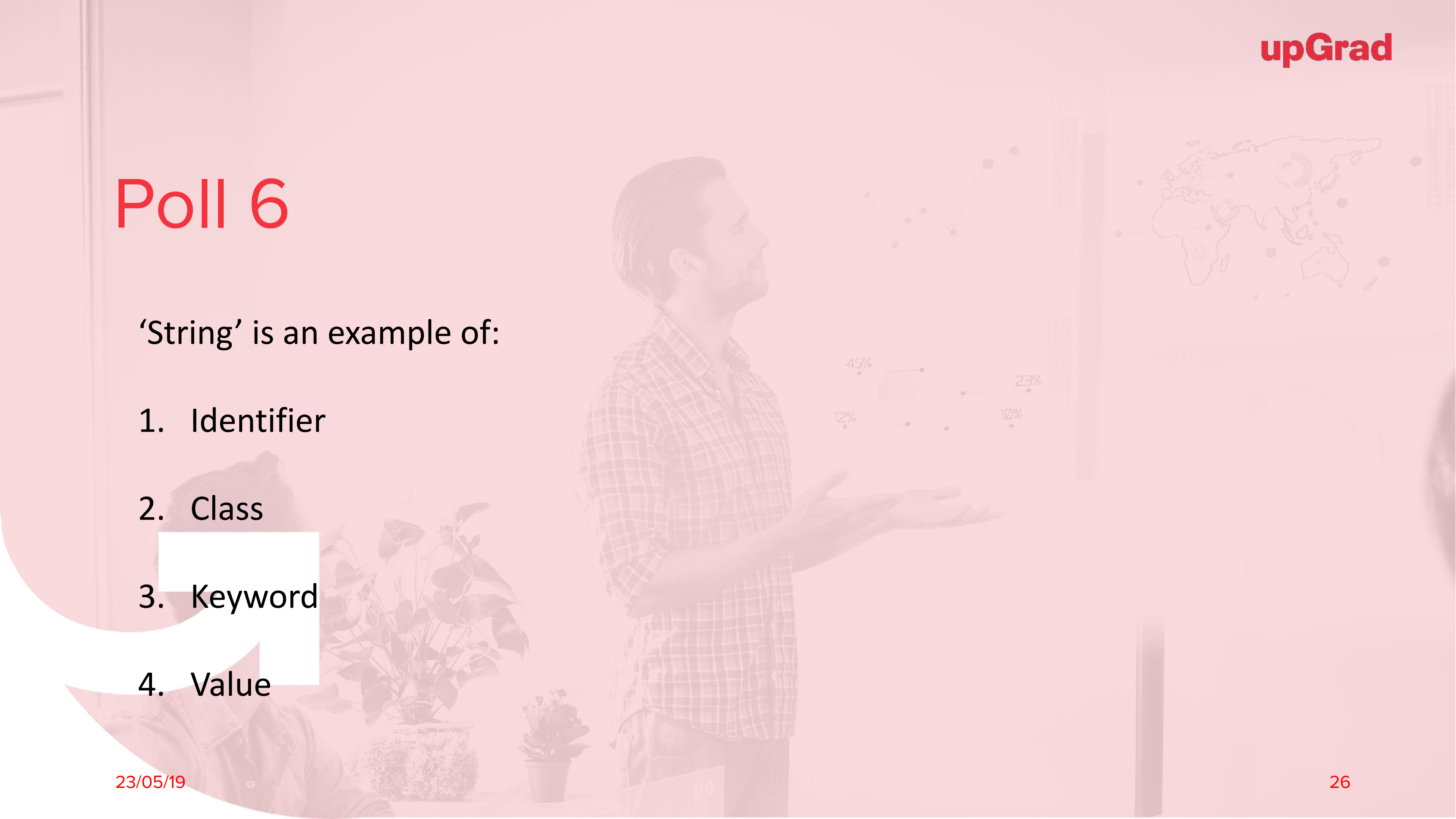
a. 123abc
b. var name
c. var_name
d. float



Poll 5 (Answer)

Which of the following is a valid variable name?

a. 123abc
b. var name
c. var_name
d. float



Poll 6

'String' is an example of:

1. Identifier
2. Class
3. Keyword
4. Value

Poll 6 (Answer)

‘String’ is an example of:

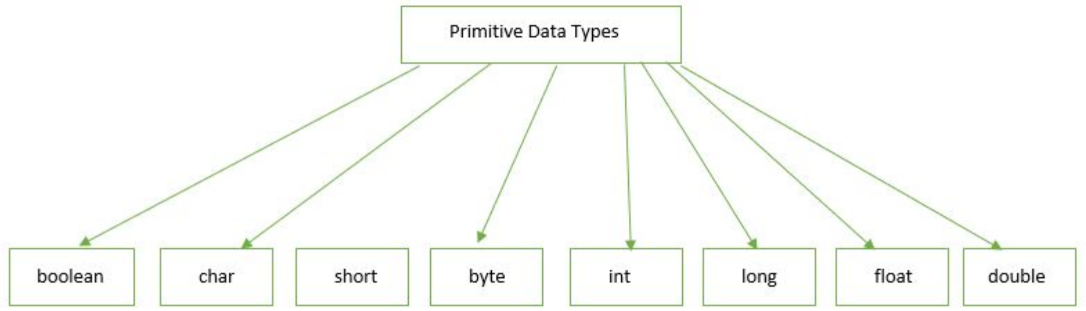
1. Identifier
2. **Class**
3. Keyword
4. Value

Data Types in Java

upGrad

- Data types tell the Java program about the type of value that will be stored in a variable.
- Java has the following two types of data types:
    - **Primitive:** int, float, char, etc.
    - **Non-primitive*:** String, array, etc.

You will learn about non-primitive data types later in the course.



Data Types in Java

upGrad

Primitive Data Types

There are eight primitive data types in Java. Let’s take a look at each of them.

1. **boolean:** The boolean data type is used to track true or false conditions. It has only two possible values: true or false. It represents one bit of information and its size is not defined precisely.

Syntax:

```
boolean boolVar;
```

Example:

```
public class Main {
  public static void main(String[] args) {
     boolean boolVar1 = true;
     boolean boolVar2 = false;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Data Types in Java

upGrad

2. byte: The byte data type is an 8-bit signed two's complement integer. It has a minimum value of –128 and a maximum value of 127 (inclusive). The byte data type can be used for saving memory in large arrays.

Syntax:

```
byte varName;
```

Example:

```
public class Main {
  public static void main(String[] args) {
     byte varName= 20;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Data Types in Java

upGrad

3. short: The short data type is a 16-bit signed two's complement integer. It has a minimum value of –32,768 and a maximum value of 32,767 (inclusive). Similar to byte, you can use a short to save memory in large arrays.

Syntax:

```
short varName;
```

Example:

```
public class Main {
  public static void main(String[] args) {
     short varName= 20;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Data Types in Java

upGrad

4. int: The int data type is a 32-bit signed two's complement integer. It has a minimum value of $–2^{31}$ and a maximum value of $2^{31}–1$.

Syntax:

```
int varName;
```

Example:

```
public class Main {
  public static void main(String[] args) {
     int varName= 20;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Data Types in Java

upGrad

5. long: The long data type is a 64-bit two's complement integer. The signed long has a minimum value of -2^{63} and a maximum value of $2^{63}-1$.

Syntax:

```
long varName;
```

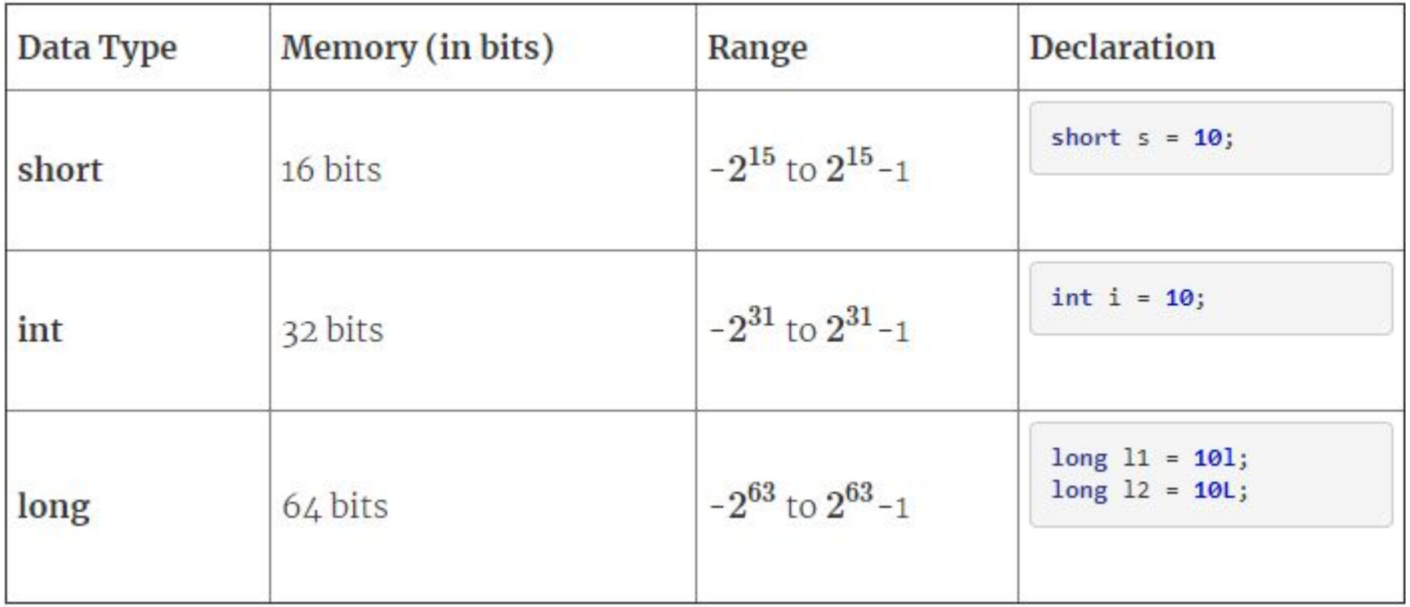
Example:

```
public class Main {
  public static void main(String[] args) {
     long varName= 20;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Comparison of Integer Data Types

upGrad



Data Type	Memory (in bits)	Range	Declaration
short	16 bits	-2^{15} to $2^{15}-1$	`short s = 10;`
int	32 bits	-2^{31} to $2^{31}-1$	`int i = 10;`
long	64 bits	-2^{63} to $2^{63}-1$	`long l1 = 10l;` `long l2 = 10L;`

Data Types in Java

upGrad

6. float: The float data type is a single-precision, 32-bit, IEEE 754 floating point.

Syntax:

```java
float varName;
```

Example:

```java
public class Main {
  public static void main(String[] args) {
     float varName= 20f;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Data Types in Java

upGrad

7. double: The double data type is a double-precision, 64-bit, IEEE 754 floating point.

Syntax:

```
double varName;
```

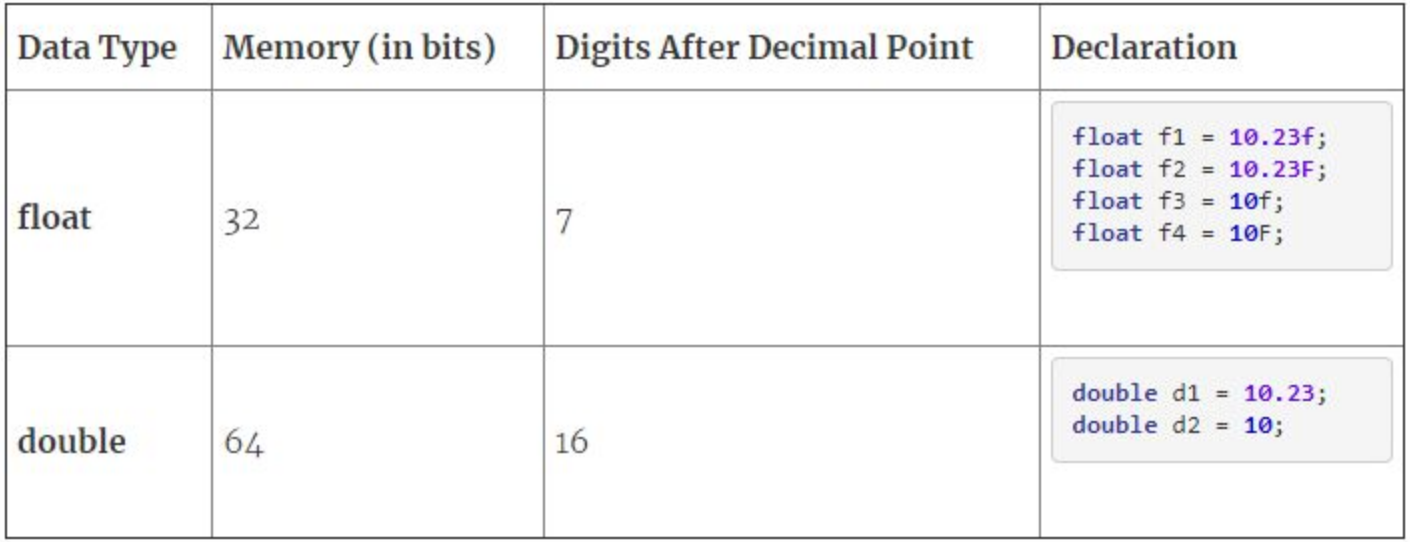
Example:

```
public class Main {
  public static void main(String[] args) {
     double varName= 20.00;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

double vs float

upGrad



Data Type	Memory (in bits)	Digits After Decimal Point	Declaration

float	32	7	`float f1 = 10.23f;`
			`float f2 = 10.23F;`
			`float f3 = 10f;`
			`float f4 = 10F;`

double	64	16	`double d1 = 10.23;`
			`double d2 = 10;`

Data Types in Java

upGrad

8. char: The char data type is a single 16-bit Unicode character. It has a minimum value of '\u0000' (or 0) and a maximum value of '\uffff' (or 65,535 inclusive).

Syntax:

```
char varName;
```

Example:

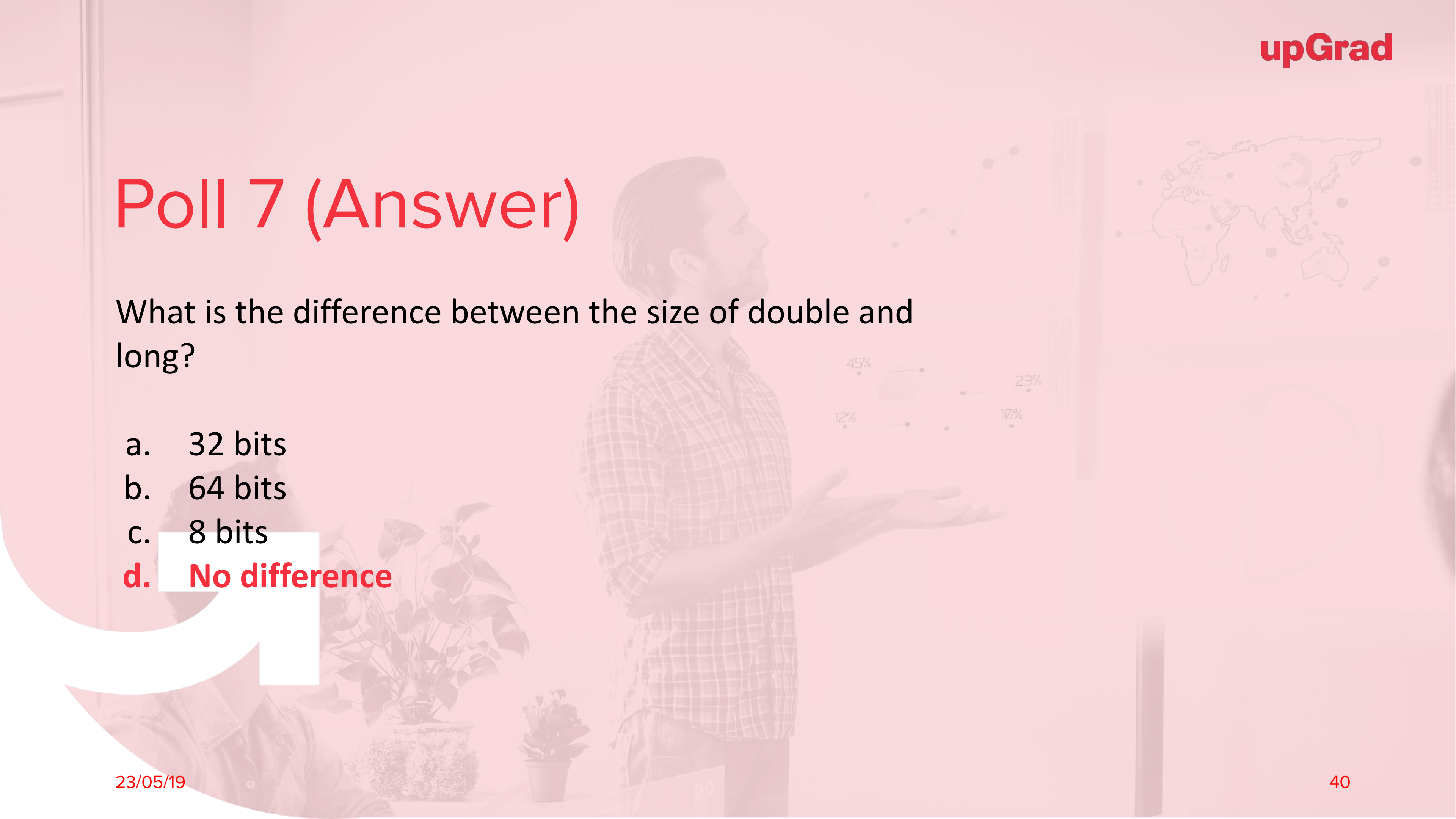
```
public class Main {
  public static void main(String[] args) {
     char varName= ’A’;
  }
}
```

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Poll 7

What is the difference between the size of double and long?

a. 32 bits
b. 64 bits
c. 8 bits
d. No difference



Poll 7 (Answer)

What is the difference between the size of double and long?

a. 32 bits
b. 64 bits
c. 8 bits
d. No difference

Hands-On Exercise – 1

upGrad

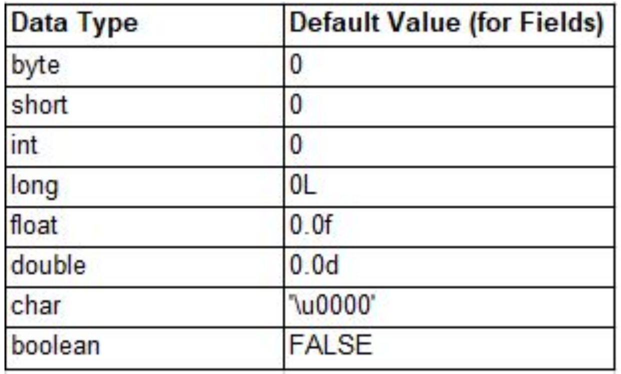
In a college, there is a need to store information on teachers as follows:

1. Name of the teacher
2. Age of the teacher
3. Gender
4. Aadhar card number
5. Address of the teacher
6. Salary up to 2 decimal places

Default Values of Data Types

upGrad

- It is not always necessary to assign a value when a field is declared.
- Fields that are declared but not initialised will be set to a default value set by the compiler.
- Variables are assigned a default value according to their data type.
- The table given below depicts the default values of the various data types.



Data Type	Default Value (for Fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	FALSE

Source: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Reading and Printing Values

upGrad

In the previous slides, you saw variable declaration and value assignment. In web applications, we often need to capture input from the user and assign it to a variable.

In Java, we use the **Scanner** class to prompt the user to input a value.

The Scanner class is used to take input from the user. It breaks the input into tokens using a delimiter pattern, which, by default, matches whitespace. The resulting tokens may then be converted to values of different types using the various methods that appear next.

To read a value from a user, you need to follow the steps given below:

1. Import the java.util.Scanner package first.
2. Create an object of the Scanner class
3. Read the input data into an variable

```java
import java.util.Scanner;

public class Main {
  public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     int num = sc.nextInt();
  }
}
```

Reading and Printing Values

upGrad

To print an output in Java, we use the System class.

The syntax of this class is as follows:

```
System.out.println(value);
```

where:

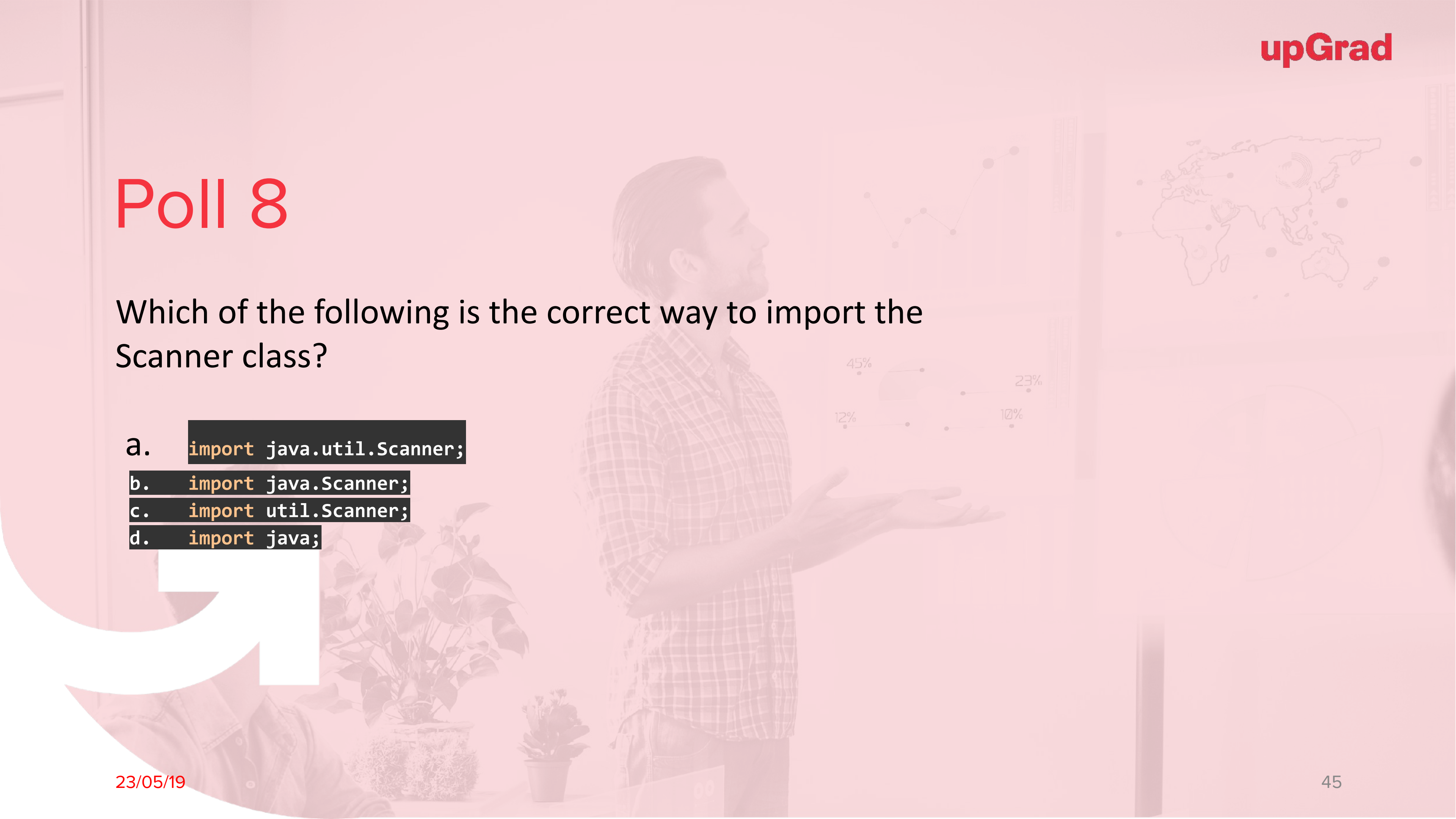
- System is a class.
- out is a field that accepts the output data.
- println is a method to print the string inside the quotes.

Note: You can have the following methods to print on the console:

1. **print():** To print the string inside the quotes
2. **println():** To print the value passed as the parameter to it on the console screen and shift the cursor to the next line on the console
3. **printf():** To print the string with formatting

```
import java.util.Scanner;

public class Main {
  public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     int num = sc.nextInt();
  }
}
```

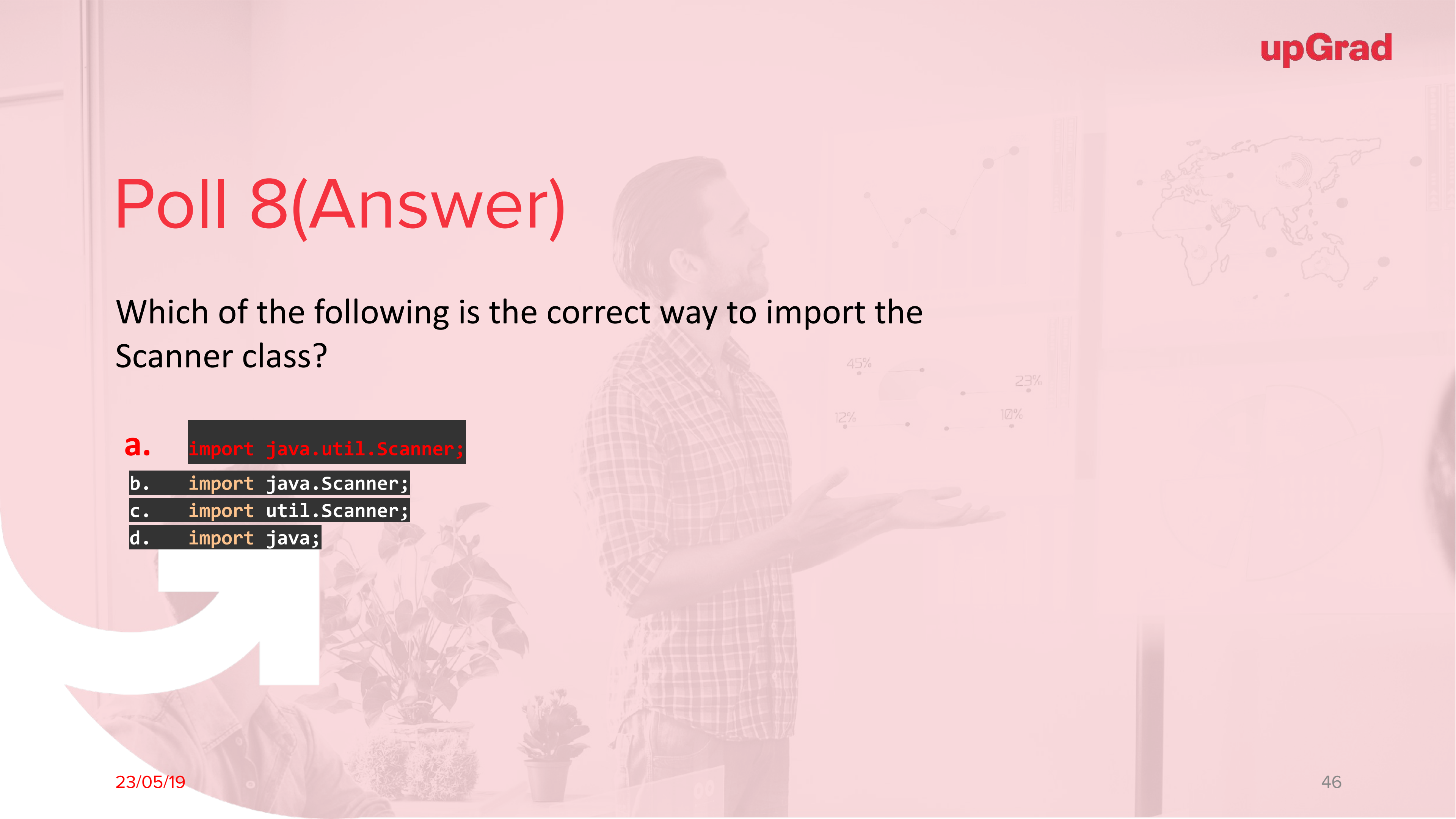


Poll 8

Which of the following is the correct way to import the Scanner class?

a. `import java.util.Scanner;`

```
b.    import java.Scanner;
c.    import util.Scanner;
d.    import java;
```



Poll 8(Answer)

Which of the following is the correct way to import the Scanner class?

a. `import java.util.Scanner;`

```
b.    import java.Scanner;
c.    import util.Scanner;
d.    import java;
```

Hands-On Exercise – 2

Get an order detail from a customer, which contains the name of the product, price and quantity. Print the values entered by the user.

Casting

Suppose you have stored the value of pi in a double variable. You want to find the integer value of pi that is stored in the same variable. How would you do this?

In programming, you will often come across situations wherein you may have to change the data type of a variable temporarily in order to perform some calculations or operations.

The process of converting one data type to another is known as casting. It is quite useful in Java programs and lets you convert one data type to another.

For example, you can convert a double into float. However, please note that you lose precision in this case.

```java
public class Main {
  public static void main(String[] args) {
     double pi = 3.141592653589793;
     System.out.println(pi);
     float floatPi = (float) pi;
     System.out.println(floatPi);
     int intPi = (int)pi;
     System.out.println(intPi);
  }
}
```

Poll 9

What would be the output of the following code?

```
float num1 = 20.14f;
double num2 = (double) num1;
System.out.println(num2);
```

a. 20.14
b. 20.14f
c. 20
d. 20.139999389648438

Poll 9(Answer)

What would be the output of the following code?

```
float num1 = 20.14f;
double num2 = (double) num1;
System.out.println(num2);
```

a. 20.14

b. 20.14f

c. 20

d. 20.139999389648438

Hands-On Exercise – 3

upGrad

Take an integer as input and cast it as a double.

Important Questions

1 What is the one basic difference between Java and all the other programming languages?

2 Can you compile a Java program in Notepad? Illustrate with an example.

3 Tell us something about Java that no one in the campus told you.

upGrad
#RahoAmbitious

Thank You!