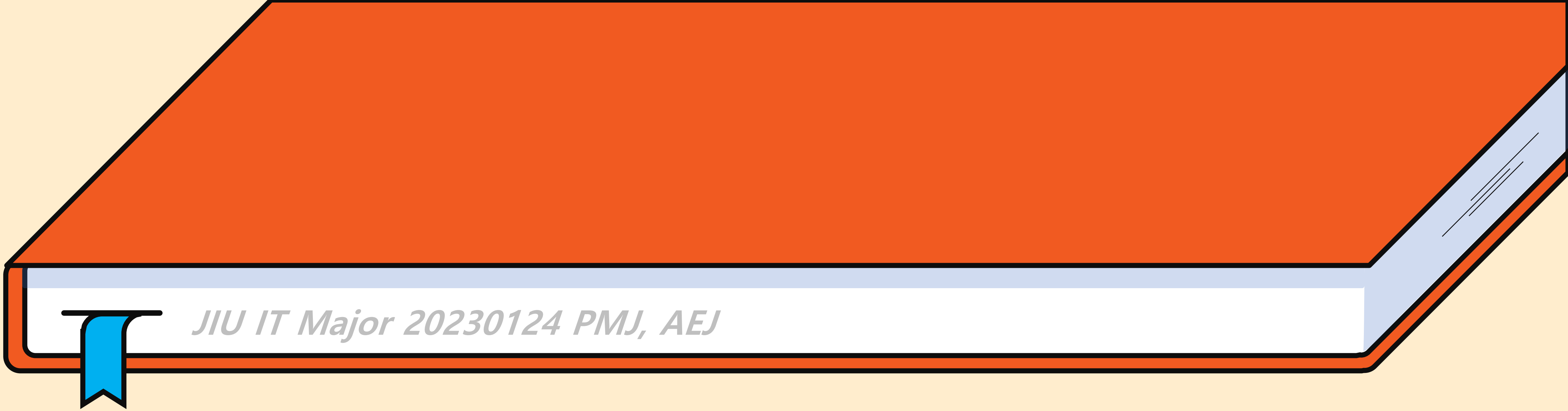


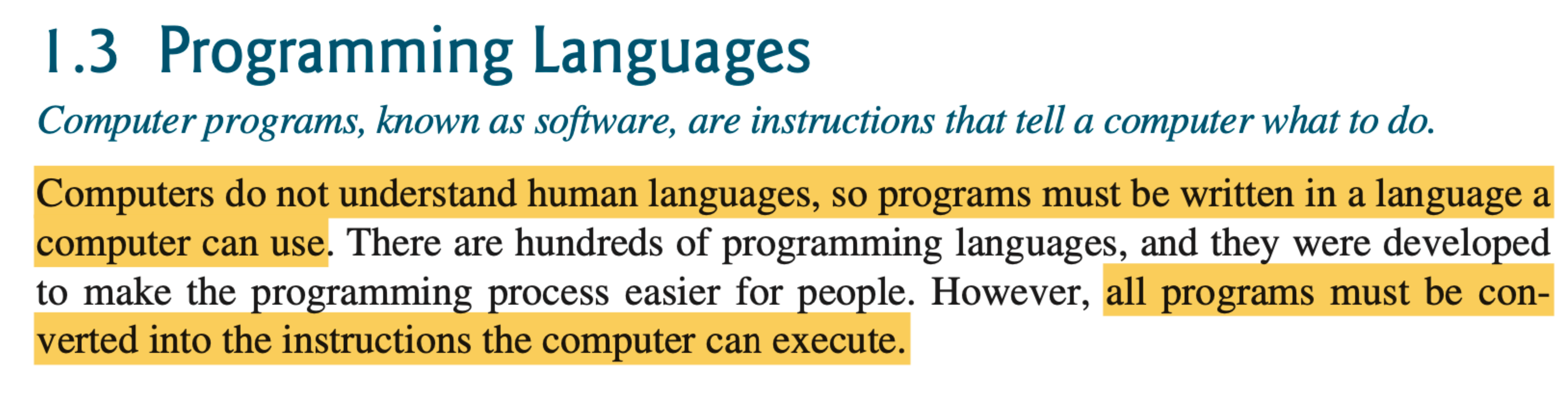
Object-Oriented Programming

Practice Week5

1.2 Computer can execute the code in ___________.

A. machine language
B. assembly language
C. high-level language
D. none of the above

1.2 Computer can execute the code in ____________.

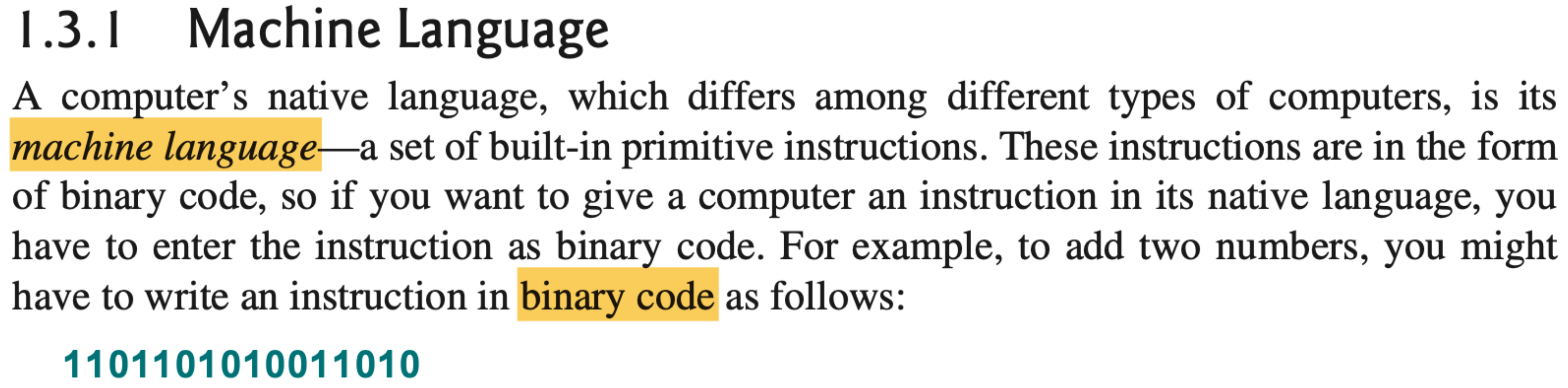


1.3 Programming Languages

Computer programs, known as software, are instructions that tell a computer what to do.

Computers do not understand human languages, so programs must be written in a language a computer can use. There are hundreds of programming languages, and they were developed to make the programming process easier for people. However, all programs must be converted into the instructions the computer can execute.

1.2 Computer can execute the code in ____________.

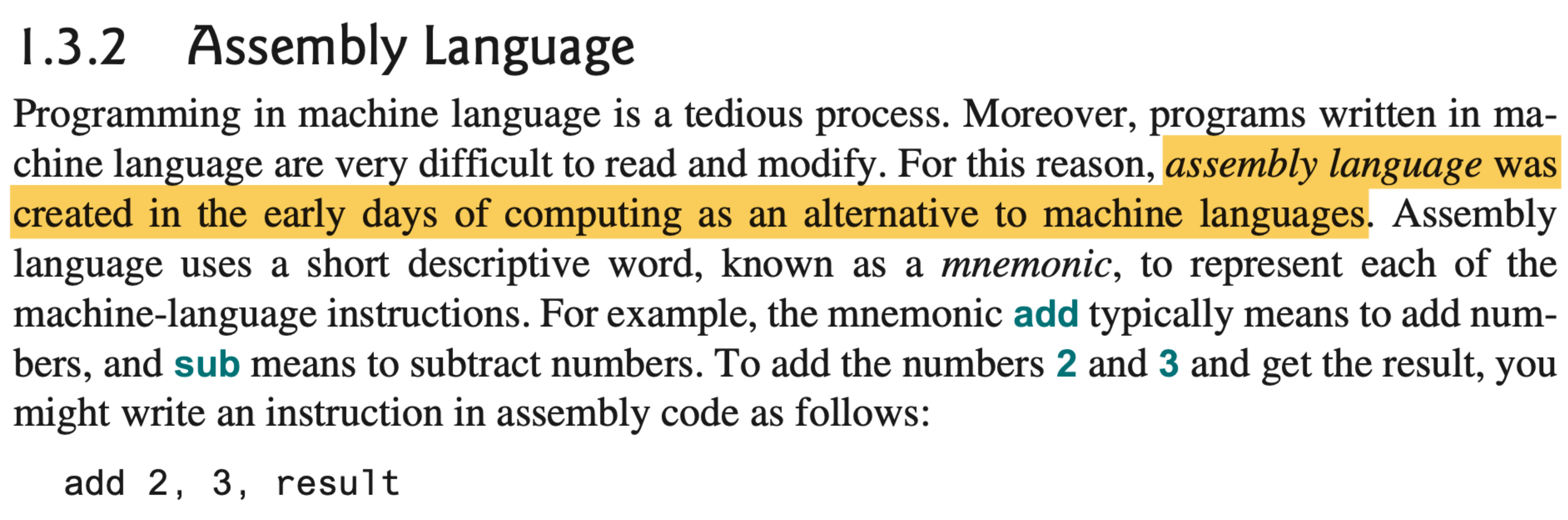


1.3.1 Machine Language

A computer's native language, which differs among different types of computers, is its *machine language*—a set of built-in primitive instructions. These instructions are in the form of binary code, so if you want to give a computer an instruction in its native language, you have to enter the instruction as binary code. For example, to add two numbers, you might have to write an instruction in binary code as follows:

```
1101101010011010
```

1.2 Computer can execute the code in _____________.

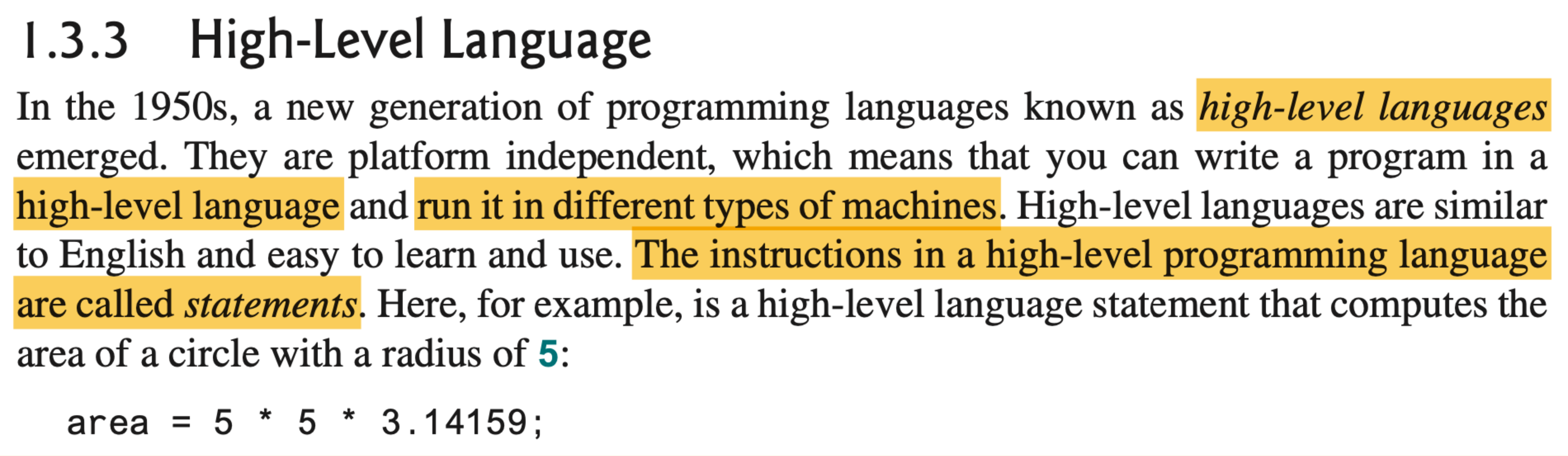


1.3.2 Assembly Language

Programming in machine language is a tedious process. Moreover, programs written in machine language are very difficult to read and modify. For this reason, *assembly language* was created in the early days of computing as an alternative to machine languages. Assembly language uses a short descriptive word, known as a *mnemonic*, to represent each of the machine-language instructions. For example, the mnemonic **add** typically means to add numbers, and **sub** means to subtract numbers. To add the numbers **2** and **3** and get the result, you might write an instruction in assembly code as follows:

```
add 2, 3, result
```

1.2 Computer can execute the code in _____________.



1.3.3 High-Level Language

In the 1950s, a new generation of programming languages known as *high-level languages* emerged. They are platform independent, which means that you can write a program in a high-level language and run it in different types of machines. High-level languages are similar to English and easy to learn and use. The instructions in a high-level programming language are called *statements*. Here, for example, is a high-level language statement that computes the area of a circle with a radius of **5**:

```
   area = 5 * 5 * 3.14159;
```

1.2 Computer can execute the code in ____________.

A. machine language
B. assembly language
C. high-level language
D. none of the above

1.3 ____________ translates high-level language program into machine language program.

A. An assembler
B. A compiler
C. CPU
D. The operating system

1.3 ____________ translates high-level language program into machine language program.

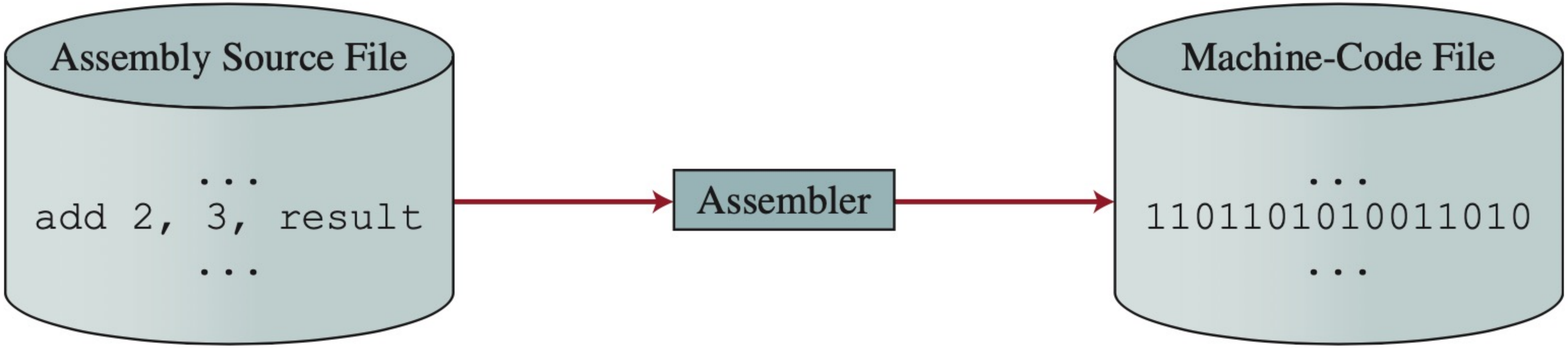
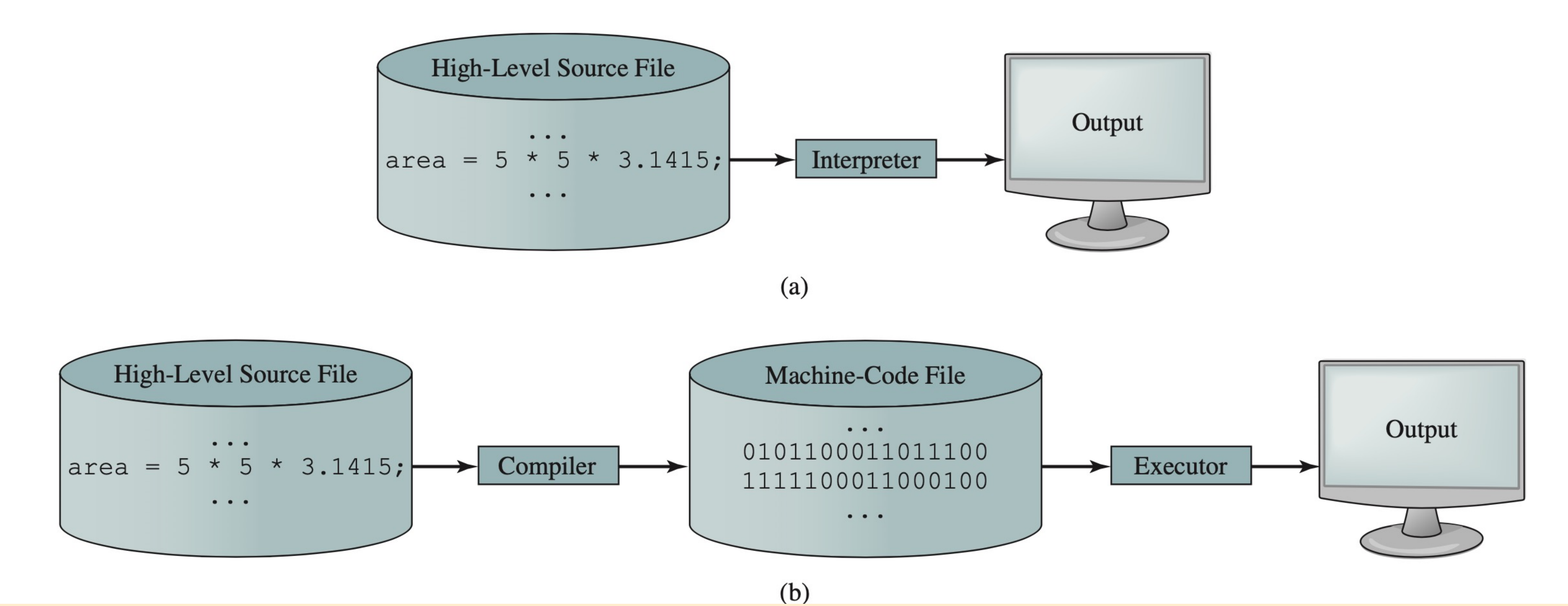


FIGURE 1.3 An assembler translates assembly-language instructions into machine code.

1.3 ____________ translates high-level language program into machine language program.



(a)

(b)

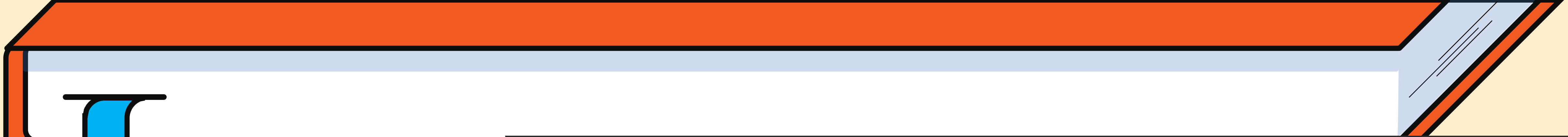
1.3 ____________ translates high-level language program into machine language program.

A. An assembler
B. A compiler
C. CPU
D. The operating system

1.7 What is the output of the following code?

```
System.out.println("1 + 2 + 3");
System.out.println(1 + 2 + 3);
```

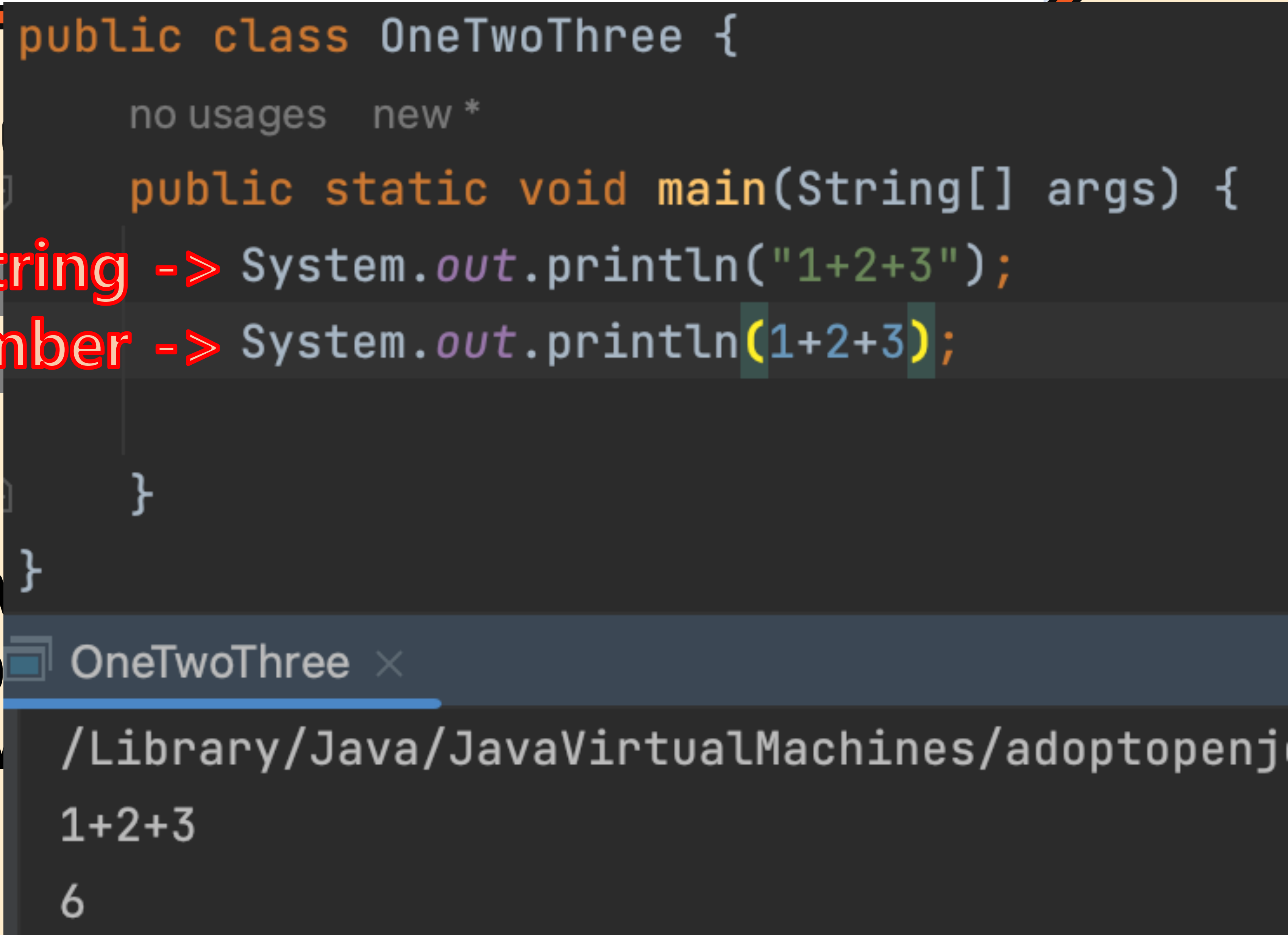
A. 1 + 2 + 3 followed by 6
B. "6" followed by 6
C. 1 + 2 + 3 followed by 1 + 2 + 3
D. 6 followed by 6



1.7 What is the o

System.out.p
System.out.p

A. 1 + 2 + 3 follow
B. "6" followed b
C. 1 + 2 + 3 follow
D. 6 followed by



String ->

Number ->

```
public class OneTwoThree {
    no usages   new *
    public static void main(String[] args) {
        System.out.println("1+2+3");
        System.out.println(1+2+3);

    }
}
```

```
/Library/Java/JavaVirtualMachines/adoptopenj
1+2+3
6
```

1.7 What is the output of the following code?

```
System.out.println("1 + 2 + 3");
System.out.println(1 + 2 + 3);
```

A. 1 + 2 + 3 followed by 6
B. "6" followed by 6
C. 1 + 2 + 3 followed by 1 + 2 + 3
D. 6 followed by 6

1.8 The JDK command to just compile a class (not run) in the file Test.java is

A.java Test

B. java Test.java

C. javac Test.java

D. javac Test

E. JAVAC Test.java

1.9 Which JDK command is correct to run a Java application in ByteCode.class?

A.java ByteCode

B. java ByteCode.class

C. javac ByteCode.java

D. javac ByteCode

E. JAVAC ByteCode

1.8 The JDK
va is

A.java Test

B.java Test.j

C. javac Test

D. javac Test

E. JAVAC Tes

1.9 Which JD
e.class?

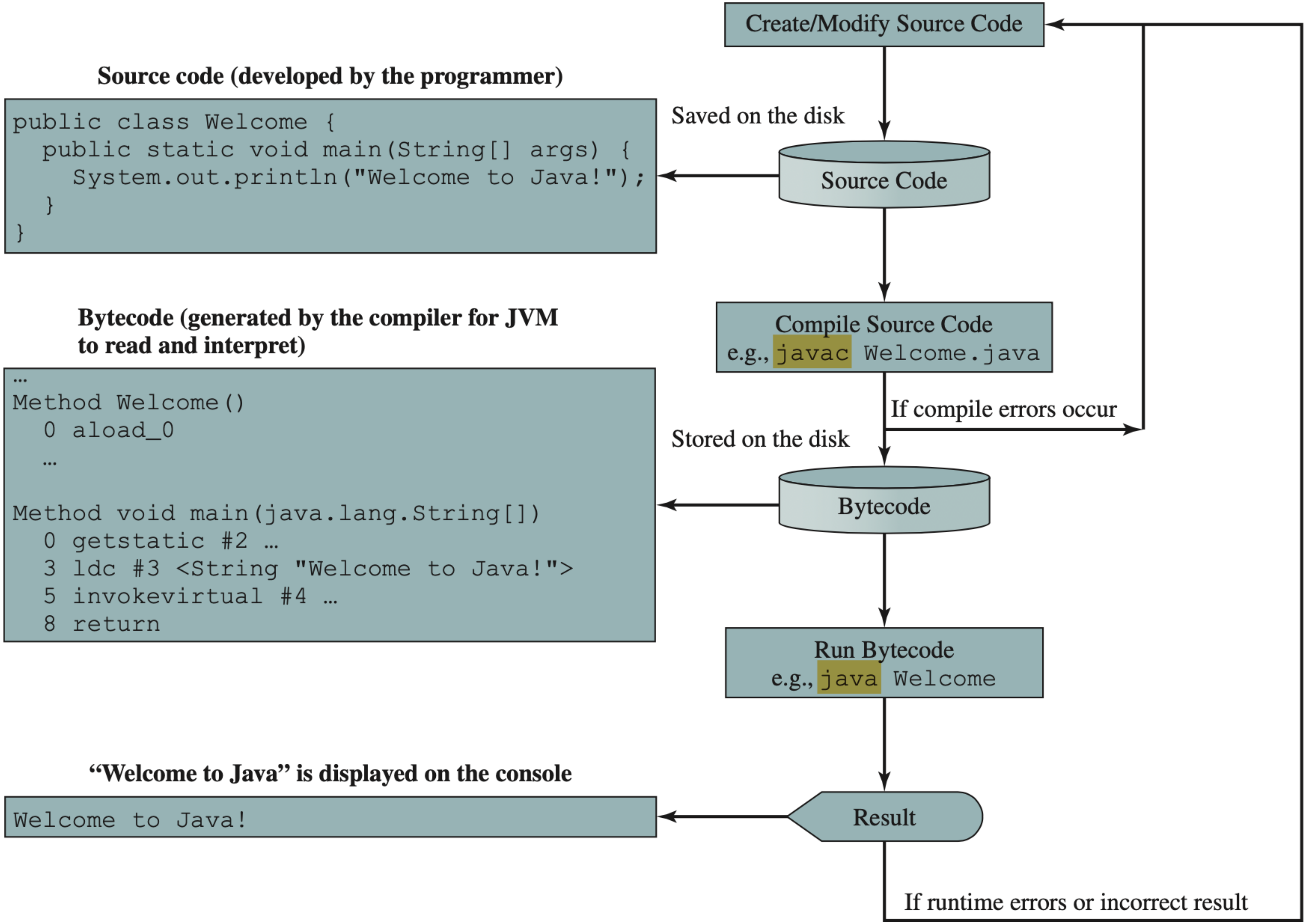
A.java ByteC

B. java ByteC

C. javac Byte

D. javac Byte

E. JAVAC Byt



Source code (developed by the programmer)

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Bytecode (generated by the compiler for JVM to read and interpret)

```
…
Method Welcome()
   0 aload_0
   …

Method void main(java.lang.String[])
   0 getstatic #2 …
   3 ldc #3 <String "Welcome to Java!">
   5 invokevirtual #4 …
   8 return
```

“Welcome to Java” is displayed on the console

```
Welcome to Java!
```

1.8 The JDK command to just compile a class (not run) in the file Test.java is

A.java Test

B.java Test.java

C. javac Test.java

D. javac Test

E. JAVAC Test.java

1.9 Which JDK command is correct to run a Java application in ByteCode.class?

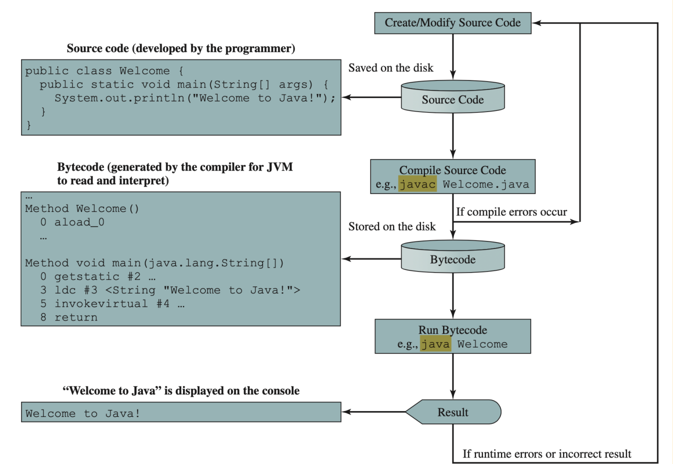
A.java ByteCode

B. java ByteCode.class

C. javac ByteCode.java

D. javac ByteCode

E. JAVAC ByteCode



Source code (developed by the programmer)

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Bytecode (generated by the compiler for JVM to read and interpret)

```
…
Method Welcome()
  0 aload_0
  …

Method void main(java.lang.String[])
  0 getstatic #2 …
  3 ldc #3 <String "Welcome to Java!">
  5 invokevirtual #4 …
  8 return
```

“Welcome to Java” is displayed on the console

```
Welcome to Java!
```

1.8 The JDK command to just compile a class (not run) in the file Test.java is

A. java Test

B. java Test.java

C. javac Test.java

D. javac Test

E. JAVAC Test.java

1.9 Which JDK command is correct to run a Java application in ByteCode.class?

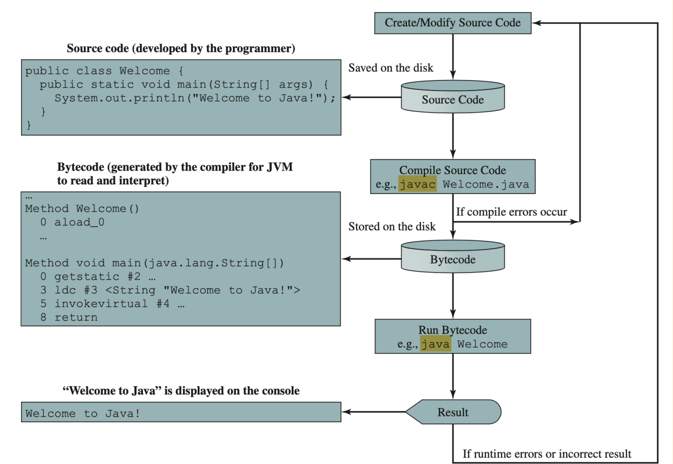
A. java ByteCode

B. java ByteCode.class

C. javac ByteCode.java

D. javac ByteCode

E. JAVAC ByteCode



Source code (developed by the programmer)

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Bytecode (generated by the compiler for JVM to read and interpret)

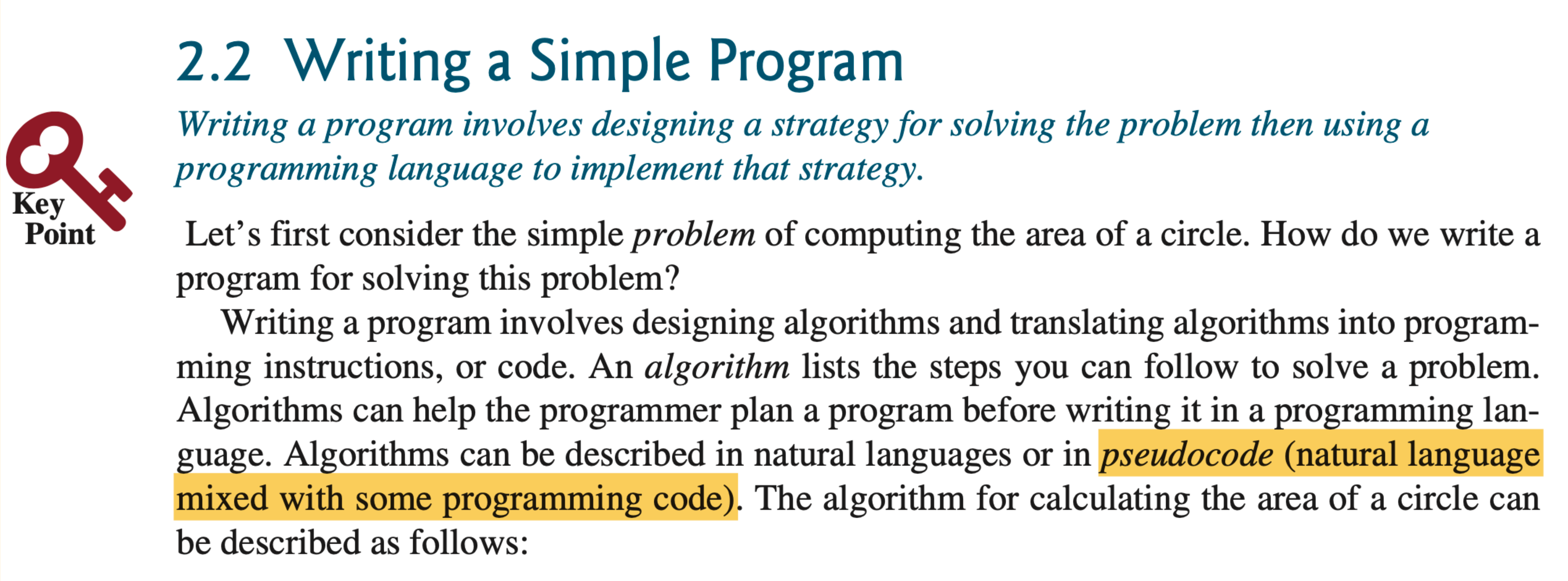
```
…
Method Welcome()
  0 aload_0
  …

Method void main(java.lang.String[])
  0 getstatic #2 …
  3 ldc #3 <String "Welcome to Java!">
  5 invokevirtual #4 …
  8 return
```

"Welcome to Java" is displayed on the console

```
Welcome to Java!
```

2.1 ________ is the code with natural language mixed with Java code.



Key Point

2.2 Writing a Simple Program

Writing a program involves designing a strategy for solving the problem then using a programming language to implement that strategy.

Let's first consider the simple *problem* of computing the area of a circle. How do we write a program for solving this problem?

Writing a program involves designing algorithms and translating algorithms into programming instructions, or code. An *algorithm* lists the steps you can follow to solve a problem. Algorithms can help the programmer plan a program before writing it in a programming language. Algorithms can be described in natural languages or in *pseudocode* (natural language mixed with some programming code). The algorithm for calculating the area of a circle can be described as follows:

2.1 _______ is the code with natural language mixed with Java code.

A. Java program
B. A Java statement
C. Pseudocode
D. A flowchart diagram

2.6 Every letter in a Java keyword is in lowercase?

A.true B. false

2.7 Which of the following is a valid identifier? Please select all that apply.

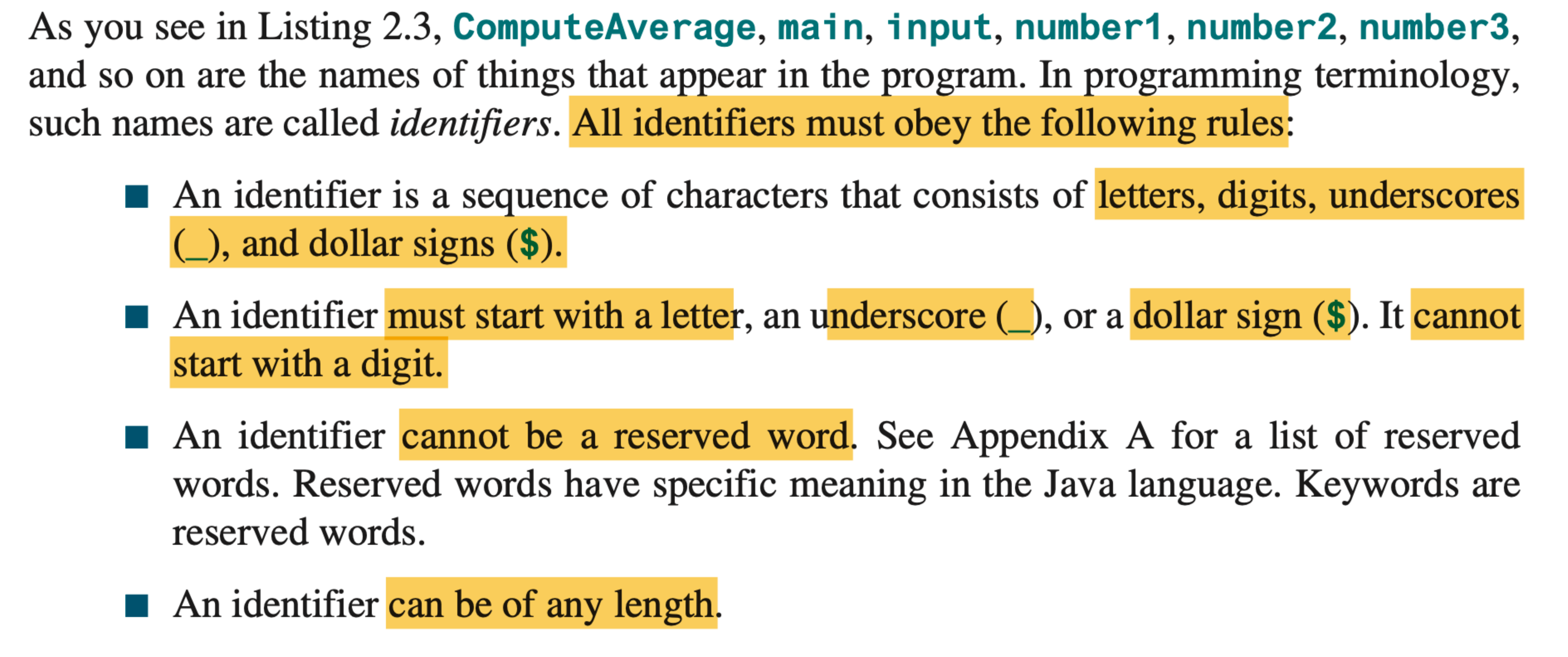
A. $343

B. class

C. 9X

D. 8+9

E. radius



As you see in Listing 2.3, `ComputeAverage`, `main`, `input`, `number1`, `number2`, `number3`, and so on are the names of things that appear in the program. In programming terminology, such names are called *identifiers*. All identifiers must obey the following rules:

- An identifier is a sequence of characters that consists of letters, digits, underscores (_), and dollar signs ($).
- An identifier must start with a letter, an underscore (_), or a dollar sign ($). It cannot start with a digit.
- An identifier cannot be a reserved word. See Appendix A for a list of reserved words. Reserved words have specific meaning in the Java language. Keywords are reserved words.
- An identifier can be of any length.

2.6 Every letter in a Java keyword is in lowercase?

A.true B. false

2.7 Which of the following is a valid identifier? Please select all that apply.

A. $343

B. class

C. 9X

D. 8+9

E. radius

2.17 Which of these data types requires the most amount of memory?

A. long B. int C. short D. byte

2.17 Which of these data types requires the most amount of memory?

A.

byte

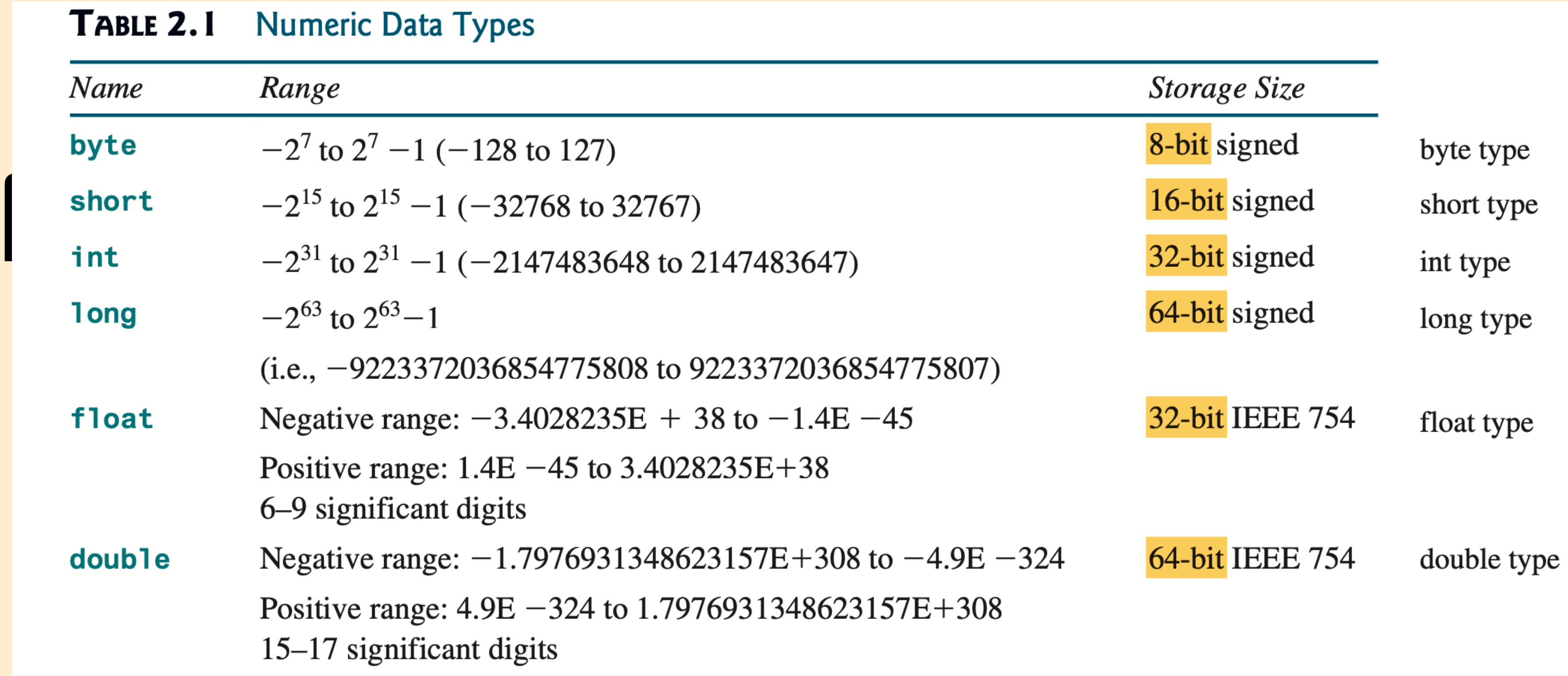


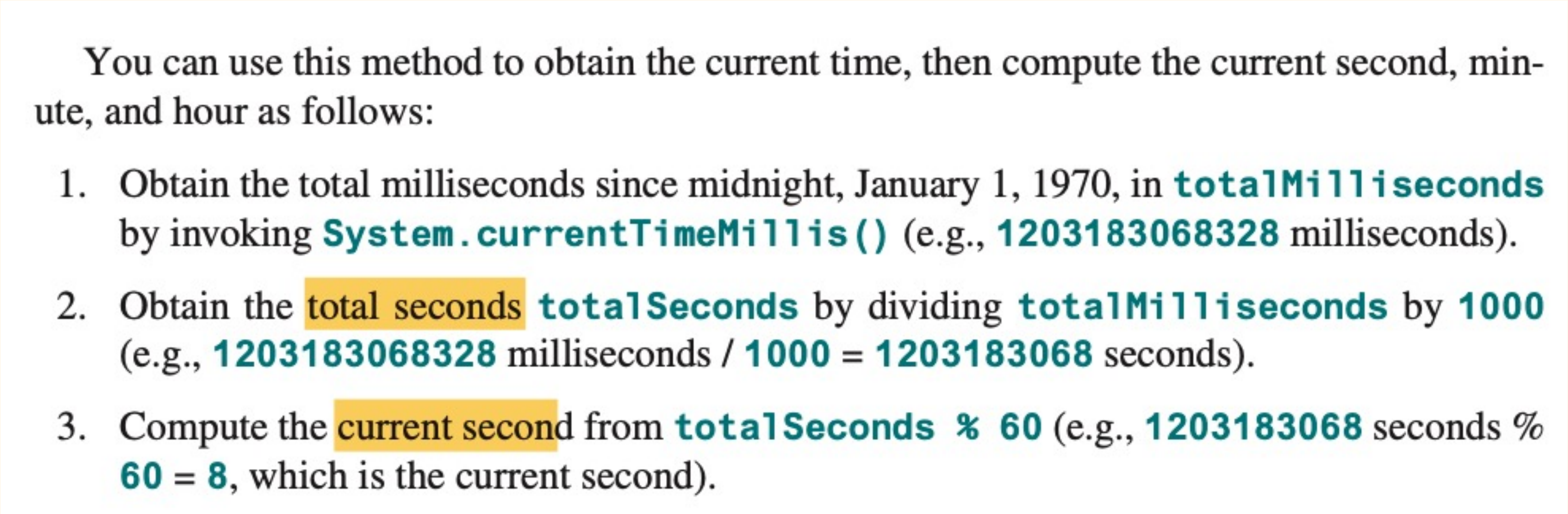
TABLE 2.1 Numeric Data Types

Name	*Range*	*Storage Size*	
`byte`	-2^7 to $2^7 -1$ (−128 to 127)	8-bit signed	byte type
`short`	-2^{15} to $2^{15} -1$ (−32768 to 32767)	16-bit signed	short type
`int`	-2^{31} to $2^{31} -1$ (−2147483648 to 2147483647)	32-bit signed	int type
`long`	-2^{63} to $2^{63}-1$ (i.e., −9223372036854775808 to 9223372036854775807)	64-bit signed	long type
`float`	Negative range: −3.4028235E + 38 to −1.4E −45 Positive range: 1.4E −45 to 3.4028235E+38 6–9 significant digits	32-bit IEEE 754	float type
`double`	Negative range: −1.7976931348623157E+308 to −4.9E −324 Positive range: 4.9E −324 to 1.7976931348623157E+308 15–17 significant digits	64-bit IEEE 754	double type

2.37 To obtain the current second, use __________.

A. System.currentTimeMillis() % 3600
B. System.currentTimeMillis() % 60
C. System.currentTimeMillis() / 1000 % 60
D. System.currentTimeMillis() / 1000 / 60 % 60
E. System.currentTimeMillis() / 1000 / 60 / 60 % 24

2.37 To obtain the current second, use _________.



You can use this method to obtain the current time, then compute the current second, minute, and hour as follows:

1. Obtain the total milliseconds since midnight, January 1, 1970, in `totalMilliseconds` by invoking `System.currentTimeMillis()` (e.g., `1203183068328` milliseconds).
2. Obtain the total seconds `totalSeconds` by dividing `totalMilliseconds` by `1000` (e.g., `1203183068328` milliseconds / `1000` = `1203183068` seconds).
3. Compute the current second from `totalSeconds % 60` (e.g., `1203183068` seconds % `60` = `8`, which is the current second).

2.37 To obtain the current second, use __________.

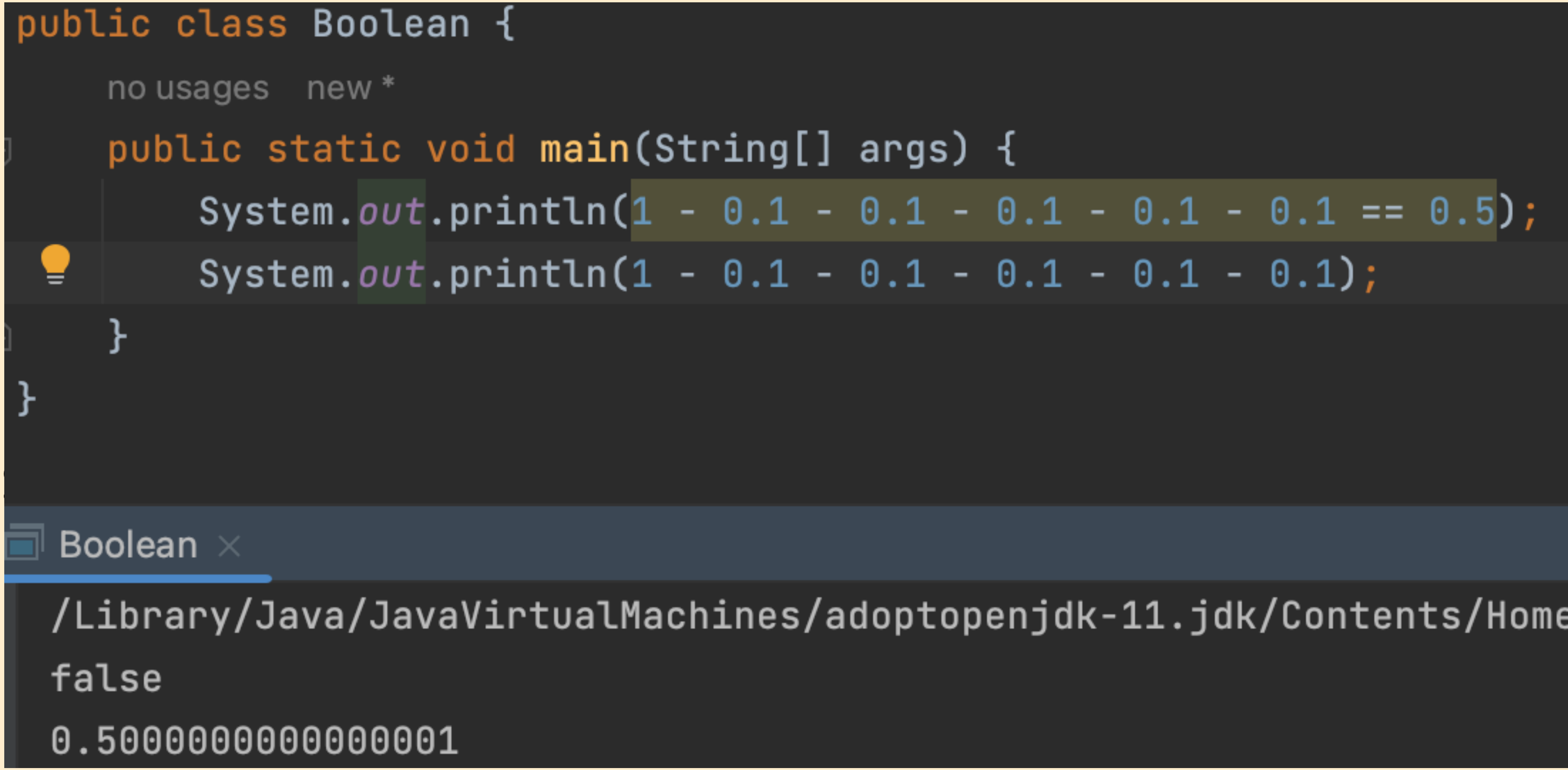
A. System.currentTimeMillis() % 3600
B. System.currentTimeMillis() % 60
C. System.currentTimeMillis() / 1000 % 60
D. System.currentTimeMillis() / 1000 / 60 % 60
E. System.currentTimeMillis() / 1000 / 60 / 60 % 24

3.4 What is 1 – 0.1 – 0.1 – 0.1 – 0.1 – 0.1 == 0.5?

A. true
B. false
C. There is no guarantee that
1 – 0.1 – 0.1 – 0.1 – 0.1 – 0.1 == 0.5 is true.

3.4 What is 1 – 0.1 – 0.1 – 0.1 – 0.1 – 0.1 == 0.5?

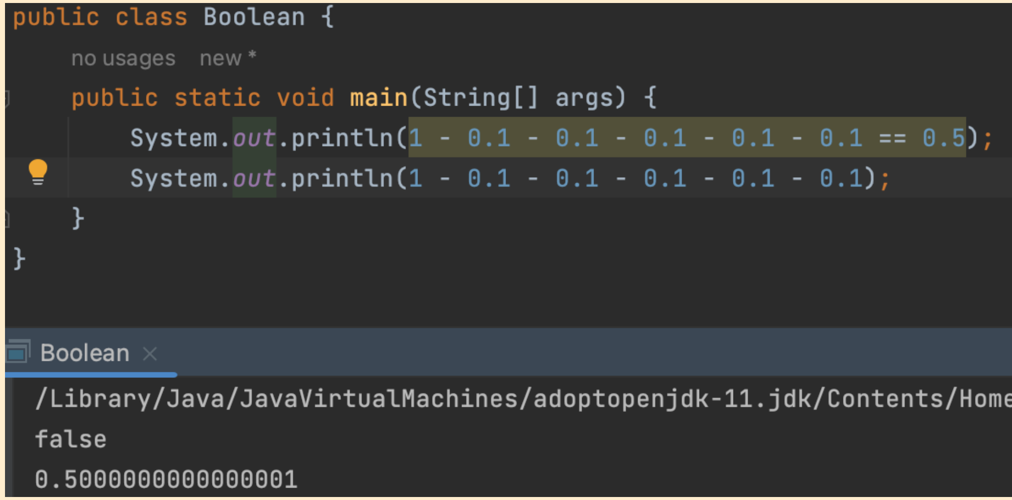
A. true
B. false
C. There i
1 – 0.1 – 0



```
public class Boolean {
    no usages   new *
    public static void main(String[] args) {
        System.out.println(1 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1 == 0.5);
        System.out.println(1 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1);
    }
}
```

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Home
false
0.5000000000000001
```

3.4 What is 1 – 0.1 – 0.1 – 0.1 – 0.1 – 0.1 == 0.5?



```
public class Boolean {
    no usages   new *
    public static void main(String[] args) {
        System.out.println(1 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1 == 0.5);
        System.out.println(1 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1);
    }
}
```

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Home
false
0.5000000000000001
```

A. true

B. false

C. There is no guarantee that 1 – 0.1 – 0.1 – 0.1 – 0.1 – 0.1 == 0.5 is true.

3.26 Given |x| <= 4, which of the following is true?

A. x <= 4 && x >= 4

B. x <= 4 && x > -4

C. x <= 4 && x >= -4

D. x <= 4 || x >= -4

3.26 Given |x| <= 4, which of the following is true?

-4 <= x <= 4

A. x <= 4 && x >= 4

B. x <= 4 && x > -4

C. x <= 4 && x >= -4

D. x <= 4 || x >= -4

3.26 Given |x| <= 4, which of the following is true?

-4 <= x <= 4

A. x <= 4 && x >= 4

B. x <= 4 && x > -4

C. x <= 4 && x >= -4

D. x <= 4 || x >= -4

3.39 What is the output of the following code?

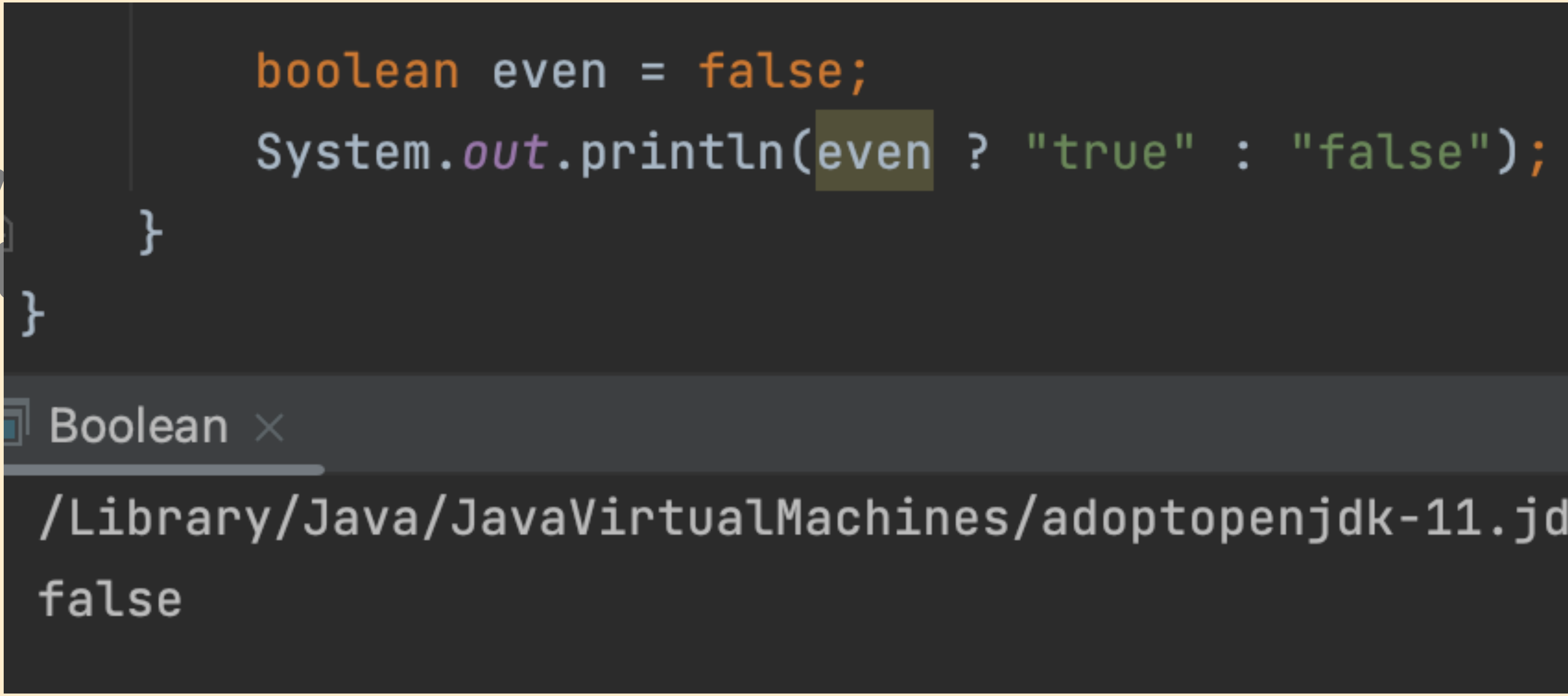
```
boolean even = false;
System.out.println(even ? "true" : "false");
```

A. true
C. nothing
B. false
D. true false

3.39 What is the output of the following code?

```
boolean ev
System.out
```

A. true
C. nothing
B. false
D. true false



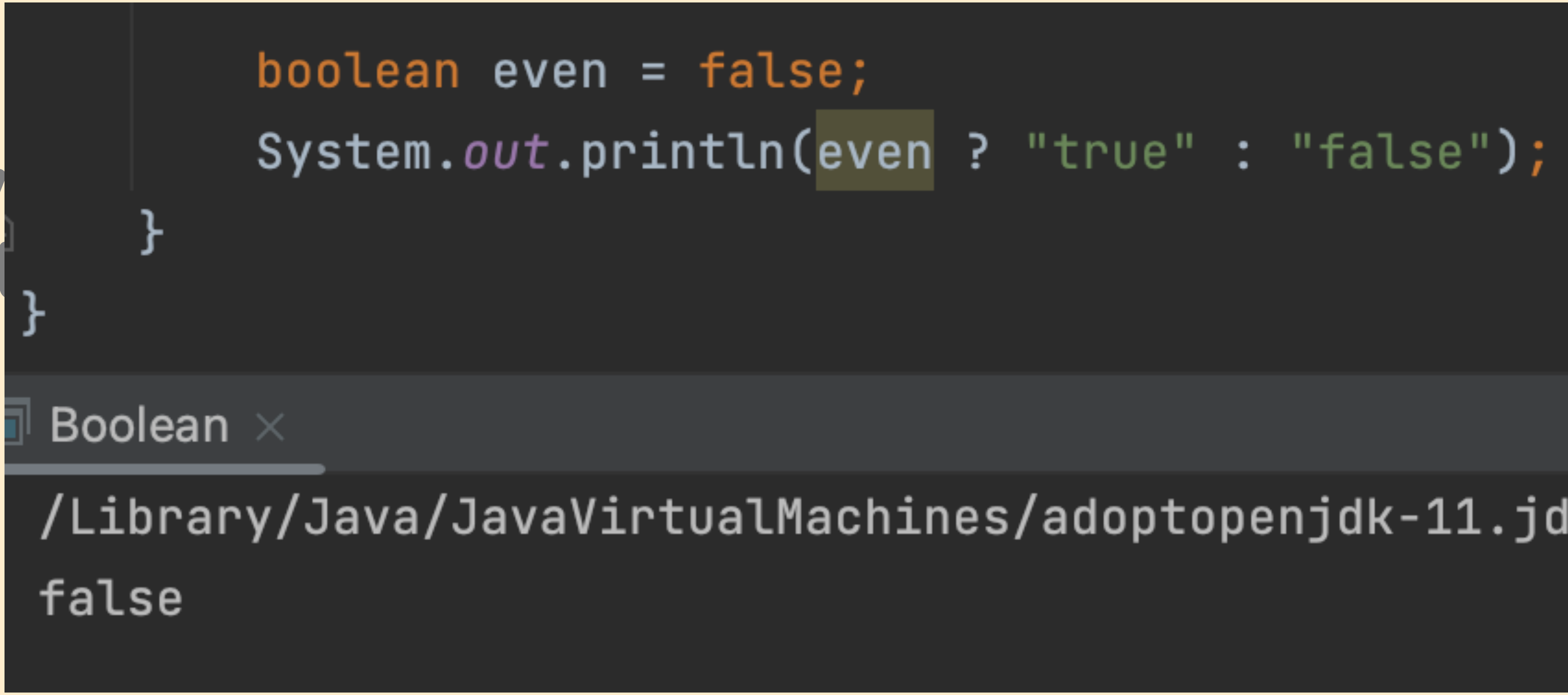
```
        boolean even = false;
        System.out.println(even ? "true" : "false");
    }
}
```

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-11.jd
false
```

3.39 What is the output of the following code?

boolean ev
System.out

A. true
C. nothing
B. false
D. true false



```
        boolean even = false;
        System.out.println(even ? "true" : "false");
    }
}
```

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-11.jd
false
```

4.1 To obtain the sine of 35 degrees, use _______.

A. Math.sin(35)
B. Math.sin(Math.toRadians(35))
C. Math.sin(Math.toDegrees(35))
D. Math.sin(Math.toRadian(35))
E. Math.sin(Math.toDegree(35))

4.1 To obtain the sine of 35 degrees, use ________.

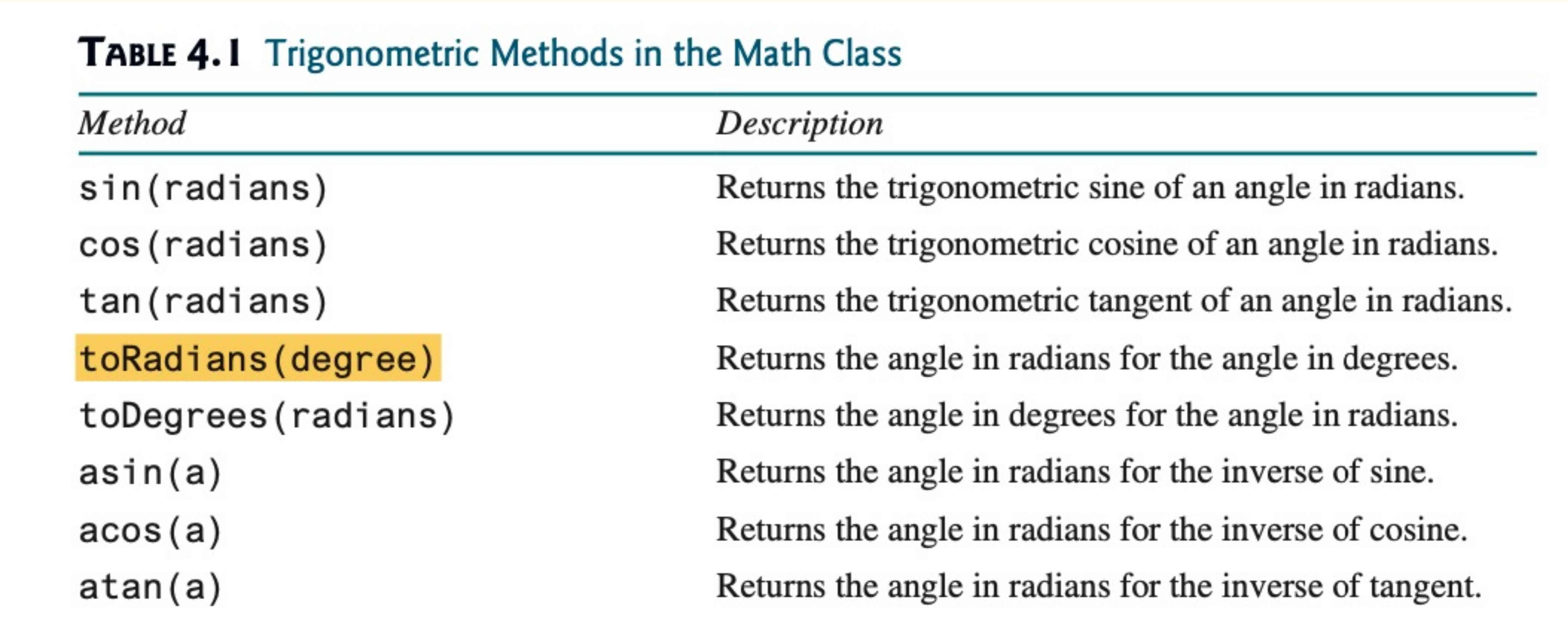


TABLE 4.1 Trigonometric Methods in the Math Class

Method	*Description*
`sin(radians)`	Returns the trigonometric sine of an angle in radians.
`cos(radians)`	Returns the trigonometric cosine of an angle in radians.
`tan(radians)`	Returns the trigonometric tangent of an angle in radians.
`toRadians(degree)`	Returns the angle in radians for the angle in degrees.
`toDegrees(radians)`	Returns the angle in degrees for the angle in radians.
`asin(a)`	Returns the angle in radians for the inverse of sine.
`acos(a)`	Returns the angle in radians for the inverse of cosine.
`atan(a)`	Returns the angle in radians for the inverse of tangent.

4.1 To obtain the sine of 35 degrees, use ________.

A. Math.sin(35)
B. Math.sin(Math.toRadians(35))
C. Math.sin(Math.toDegrees(35))
D. Math.sin(Math.toRadian(35))
E. Math.sin(Math.toDegree(35))

4.18 Suppose x is a char variable with a value 'b’. What is the output of the statement

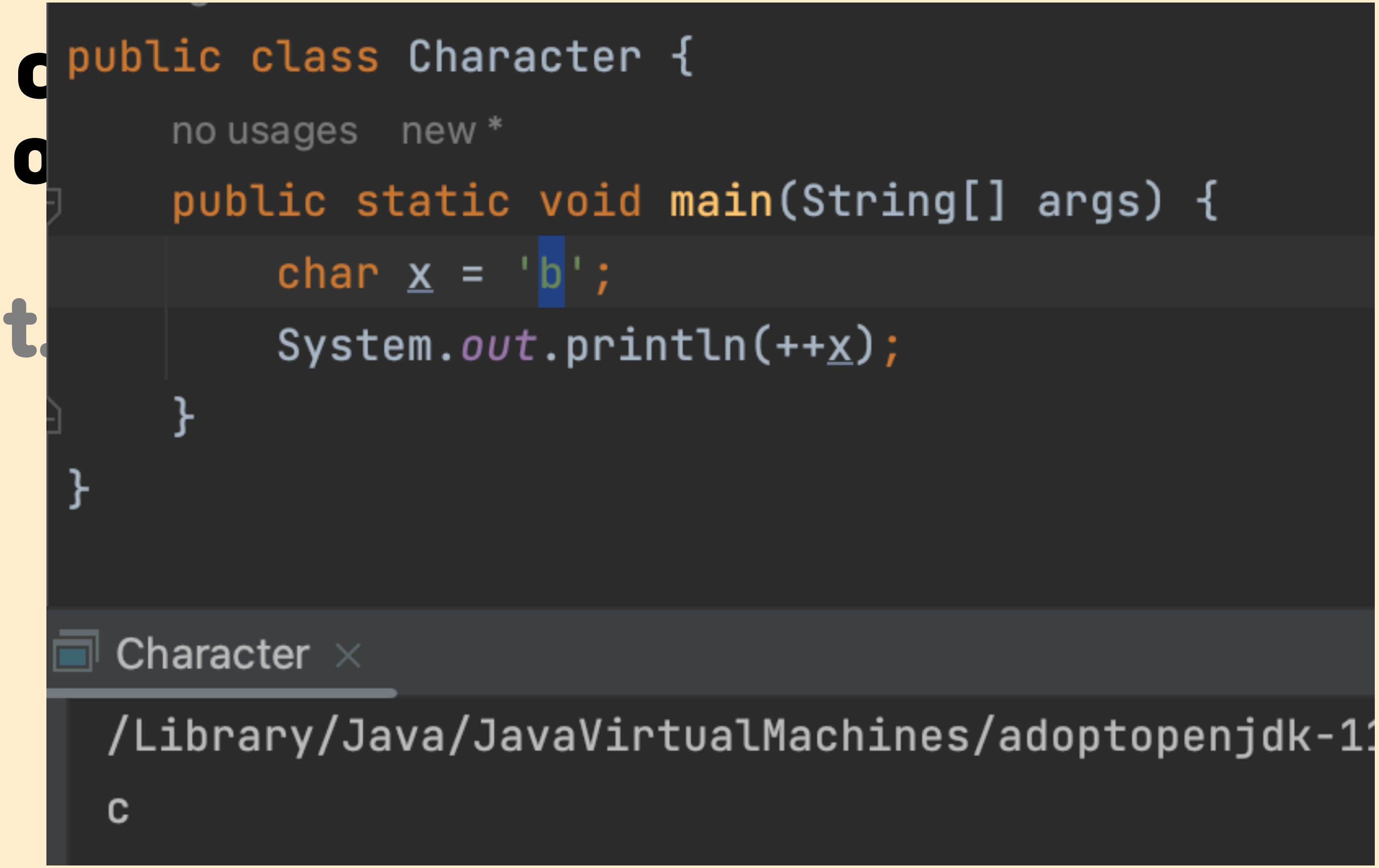
```
System.out.println(++x)?
```

A. a B. b C. c D. d

4.18 Suppose x is a c
What is the output o

System.out.

A. a B. b C. c D. d



```
public class Character {
    public static void main(String[] args) {
        char x = 'b';
        System.out.println(++x);
    }
}
```

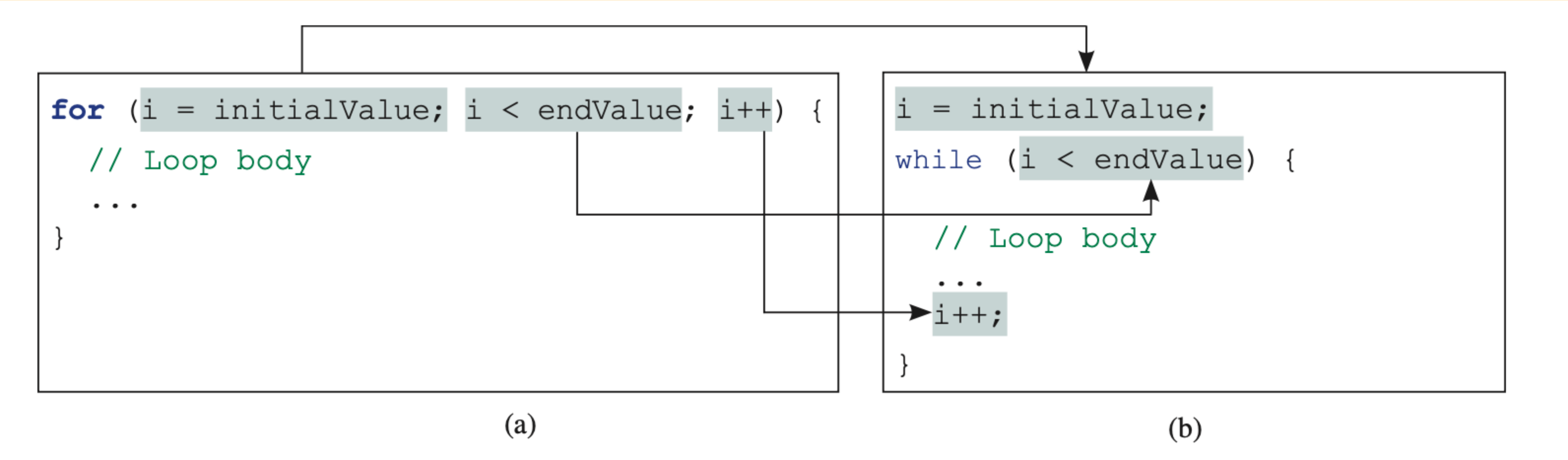
```
/Library/Java/JavaVirtualMachines/adoptopenjdk-1
c
```

4.18 Suppose x is a char variable with a value 'b’. What is the output of the statement

```
System.out.println(++x)?
```

A. a B. b [✓ C. c] D. d

Difference between for and while



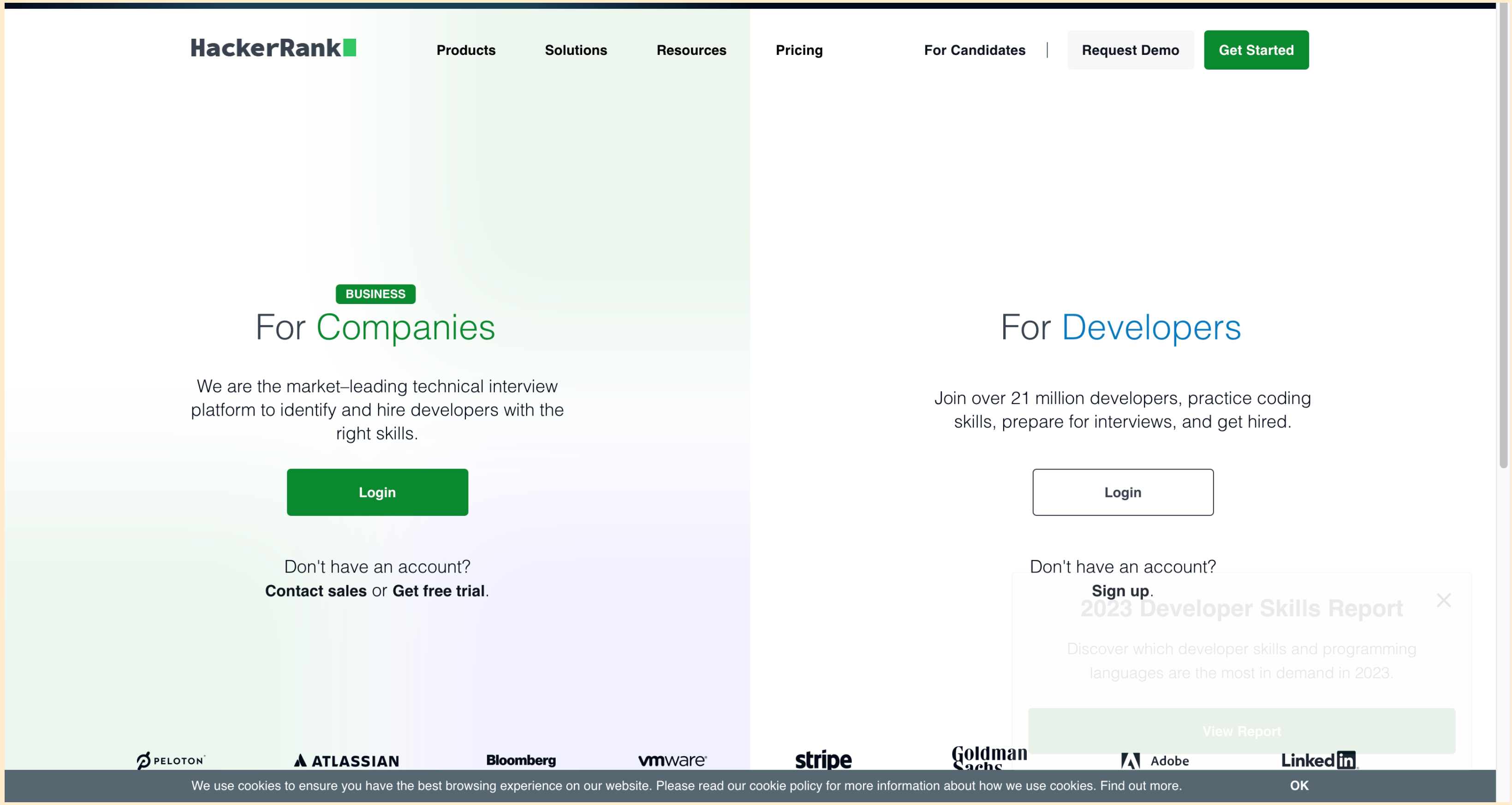
```
for (i = initialValue; i < endValue; i++) {
  // Loop body
  ...
}
```

(a)

```
i = initialValue;
while (i < endValue) {
  // Loop body
  ...
  i++;
}
```

(b)

https://www.hackerrank.com/



BUSINESS

For Companies

We are the market–leading technical interview platform to identify and hire developers with the right skills.

Login

Don't have an account?
Contact sales or **Get free trial**.

For Developers

Join over 21 million developers, practice coding skills, prepare for interviews, and get hired.

Login

Don't have an account?
Sign up.

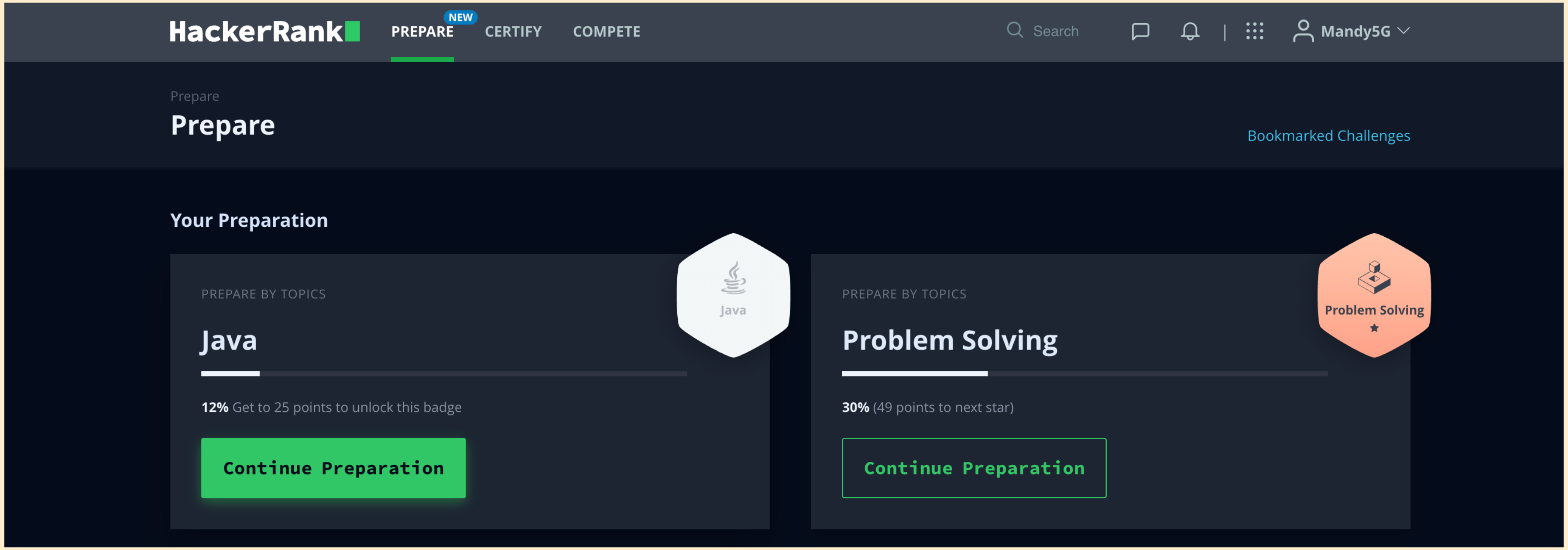
2023 Developer Skills Report

Discover which developer skills and programming languages are the most in demand in 2023.

View Report

We use cookies to ensure you have the best browsing experience on our website. Please read our cookie policy for more information about how we use cookies. Find out more.

OK



Prepare

Prepare

Bookmarked Challenges

Your Preparation

PREPARE BY TOPICS

Java

12% Get to 25 points to unlock this badge

Continue Preparation

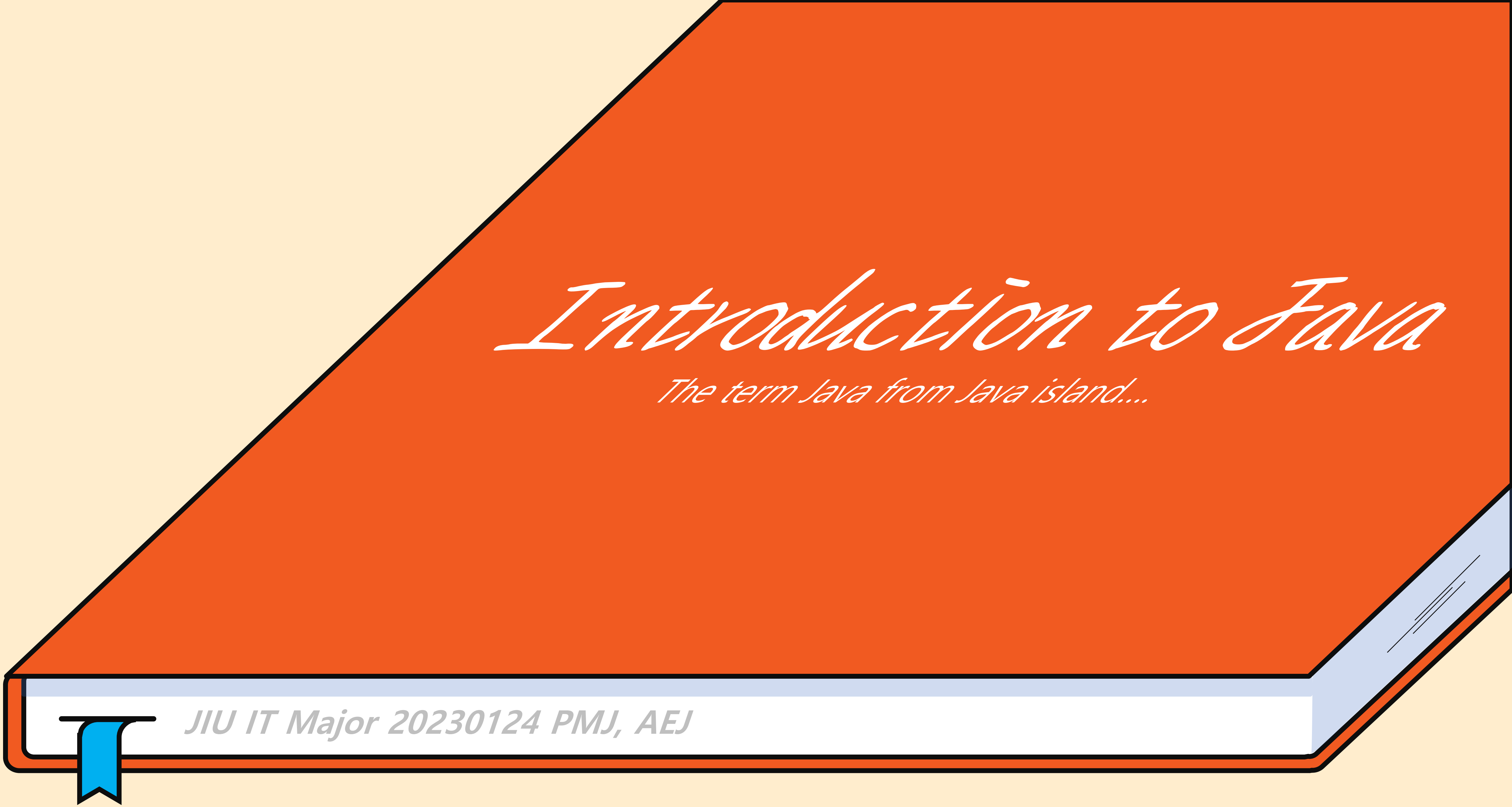
PREPARE BY TOPICS

Problem Solving

30% (49 points to next star)

Continue Preparation

https://www.hackerrank.com/



Introduction to Java

The term Java from Java island...

JIU IT Major 20230124 PMJ, AEJ