

# Shubham Raut

## Java Assignment - I

---

Q. 1. Download and install oracle JDK on your machine and explore JDK home & JRE home directory.

Q 2. Copy src.zip and rt.jar on desktop. Extract them and observe the directories as well as files & their extensions.

```
proxim@Proxims-MacBook-Air JavaVirtualMachines % tree /Library/Java/JavaVirtualMachines
/Library/Java/JavaVirtualMachines
└── jdk-22.0.2.jdk
    ├── Contents
    │   ├── CodeResources
    │   ├── Home
    │   │   ├── LICENSE -> legal/java.base/LICENSE
    │   │   ├── README
    │   │   └── bin
    │   │       ├── jar
    │   │       ├── jarsigner
    │   │       ├── java
    │   │       ├── javac
    │   │       ├── javadoc
    │   │       ├── javap
    │   │       ├── jcmd
    │   │       ├── jconsole
    │   │       ├── jdb
    │   │       ├── jdepscan
    │   │       ├── jdeps
    │   │       ├── jfr
    │   │       ├── jhsdb
    │   │       ├── jimage
    │   │       ├── jinfo
    │   │       ├── jlink
    │   │       ├── jmap
    │   │       ├── jmod
    │   │       ├── jpackage
    │   │       ├── jps
    │   │       ├── jrunscript
    │   │       ├── jshell
    │   │       ├── jstack
    │   │       ├── jstat
    │   │       ├── jstard
    │   │       ├── jwebserver
    │   │       ├── keytool
    │   │       ├── rmiregistry
    │   │       └── serialver
    │   └── conf
        ├── jaxp.properties
        ├── logging.properties
        ├── management
        │   ├── jmxremote.access
        │   ├── jmxremote.password.template
        │   └── management.properties
        ├── net.properties
        ├── security
        │   ├── java.policy
        │   ├── java.security
        │   └── policy
        │       ├── README.txt
        │       └── limited
```

Q3. Write a simple "Hello World!" application in any text editor and compile & run it from terminal.



```
1 // Q3 .
2 public class Helloworld {
3     public static void main(String[] args){
4         System.out.println("Hello World!");
5     }
6 }
7 }
8
9
```

Q4. Set path permanently in environment variable and test "Hello World!" application again.

```
▶ proxim@Proxims-MacBook-Air 2.java % echo $JAVA_HOME
which java
/Users/proxim/.jenv/versions/temurin64-24.0.2
/Users/proxim/.jenv/shims/java
```

```
proxim@Proxims-MacBook-Air 2.java % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/2.java/assignments/assignment-1/" && javac HelloWorld.java && java HelloWorld
Hello World!
```

Q5. Use Java disassembler and its switches to observe bytecode.

```
proxim@Proxims-MacBook-Air assignment-1 % javap HelloWorld
Compiled from "HelloWorld.java"
public class HelloWorld {
    public HelloWorld();
    public static void main(java.lang.String[]);
}
```

Q6. Write a program to perform below operations on Boolean type to convert:

- boolean value into String

```
assignments > assignment-1 > BooleanToString.java > BooleanToString
1  public class BooleanToString{
    Run | Debug
2      public static void main(String[] args) {
3          boolean isLightOn = true;
4          String lightStatus = String.valueOf(isLightOn);
5          System.out.println("Light"+lightStatus);
6      }
7 }
```

b. boolean value into Boolean instance.

```
public class Boolean_instance{
    Run | Debug
    public static void main(String[] args) {
        boolean isLightOn = true;
        Boolean lightObj = Boolean.valueOf(isLightOn);
        System.out.println(lightObj);
    }
}
```

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/2.java/assignments/assignment-1/" && javac Boolean_instance.java && java Boolean_instance
true
```

c. String value into boolean value

```
assignments > assignment-1 > 🚧 Parse_boolean.java > 🛡 Parse_boolean > ⚙ main(String[])
 1  //6. b string value into boolean value
 2  // "true" (in any case) are allowed in (converted to true).
 3  // Everyone else - whether they say "zebra", "hello", "yes", "1",
 4  // or anything else - gets turned away (converted to false).
 5  public class Parse_boolean {
    Run | Debug
 6  |  public static void main(String args[]){
 7  |  |  String door = "true";
 8  |  |  boolean isDoorOpen = Boolean.parseBoolean(door);
 9  |  |  System.out.println(door);
10  |  |  System.out.println(isDoorOpen);
11  |
12  |
13  |  |  } [ ]
14  |  }
15  |
16  |
```

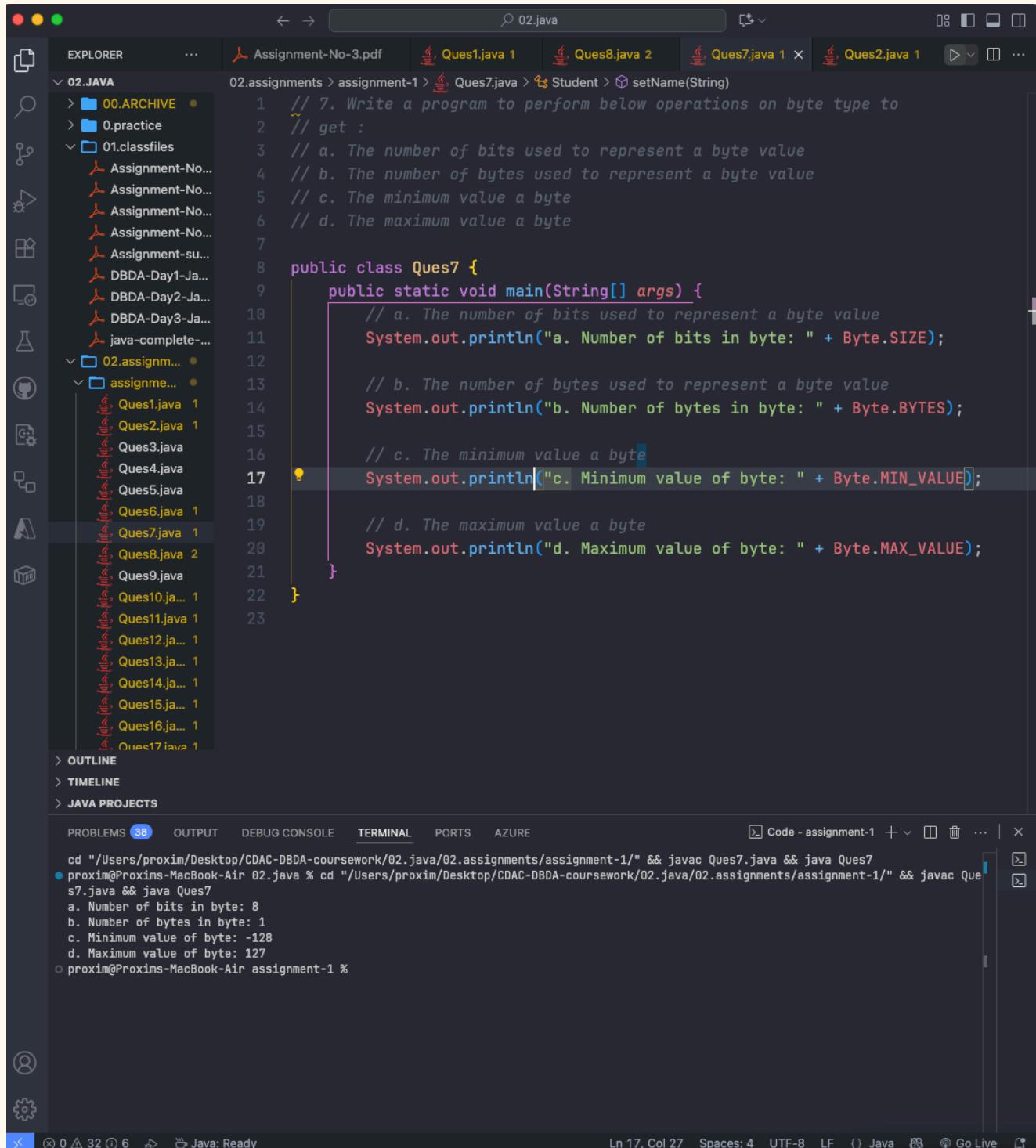
```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/2.java/assignments/assignment-1/" && javac Parse_boolean.java && java Parse_boolean
true
```

d. String value into Boolean instance.

```
assignments > assignment-1 > 🚧 String_Bool_instance.java > 🛡 String_Bool_instance > ⚙ main(String[])
 1  public class String_Bool_instance {
    Run | Debug
 2  |  public static void main(String[] args){
 3  |  |  String hello = "false";
 4  |  |  Boolean result = Boolean.valueOf(hello);
 5  |  |  System.out.println(result);
 6  |
 7  |  |  } [ ]
 8  |
 9  }
10 |
```

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/2.java/assignments/assignment-1/" && javac String_Bool_instance.java && java String_Bool_instance
true
```

## Ques 7.



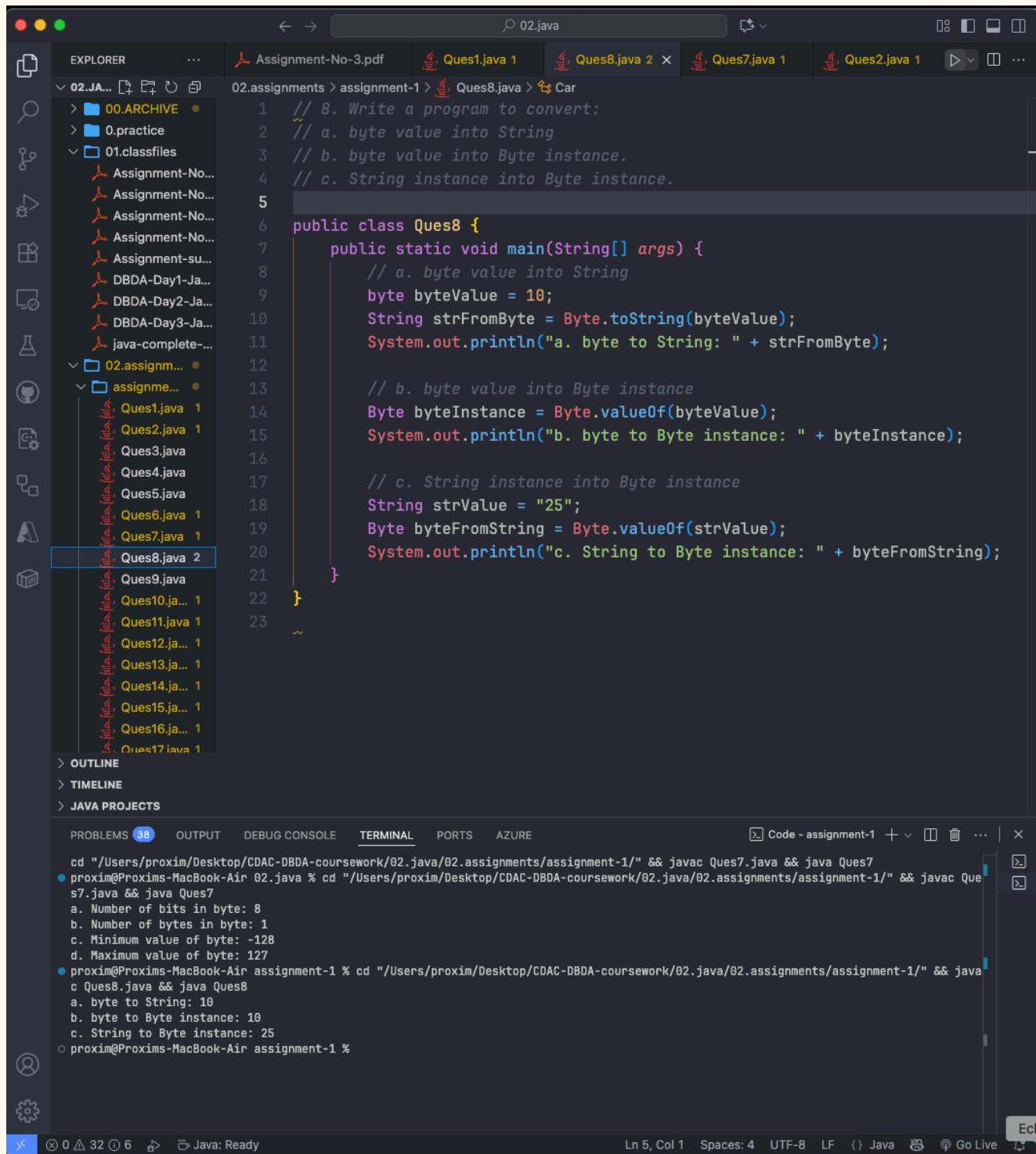
The screenshot shows a Java code editor in VS Code. The code is a program to perform operations on byte type. The code is as follows:

```
1 // 7. Write a program to perform below operations on byte type to
2 // get :
3 // a. The number of bits used to represent a byte value
4 // b. The number of bytes used to represent a byte value
5 // c. The minimum value a byte
6 // d. The maximum value a byte
7
8 public class Ques7 {
9     public static void main(String[] args) {
10         // a. The number of bits used to represent a byte value
11         System.out.println("a. Number of bits in byte: " + Byte.SIZE);
12
13         // b. The number of bytes used to represent a byte value
14         System.out.println("b. Number of bytes in byte: " + Byte.BYTES);
15
16         // c. The minimum value a byte
17         System.out.println("c. Minimum value of byte: " + Byte.MIN_VALUE);
18
19         // d. The maximum value a byte
20         System.out.println("d. Maximum value of byte: " + Byte.MAX_VALUE);
21     }
22 }
23 }
```

The code editor has a sidebar with a file tree showing various Java files (Ques1.java 1, Ques2.java 1, Ques3.java, Ques4.java, Ques5.java, Ques6.java 1, Ques7.java 1, Ques8.java 2, Ques9.java, Ques10.java 1, Ques11.java 1, Ques12.java 1, Ques13.java 1, Ques14.java 1, Ques15.java 1, Ques16.java 1, Ques17.java 1) and a main file 02.java. The terminal at the bottom shows the output of running the program:

```
cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && javac Ques7.java && java Ques7.java && java Ques7
a. Number of bits in byte: 8
b. Number of bytes in byte: 1
c. Minimum value of byte: -128
d. Maximum value of byte: 127
```

## Ques 8



The screenshot shows a Java code editor in an IDE. The code is for a class named Ques8, which contains a main method. The code is intended to convert a byte value into a String, a Byte instance, and a String instance into a Byte instance. The code uses the Byte class's static methods to perform these conversions.

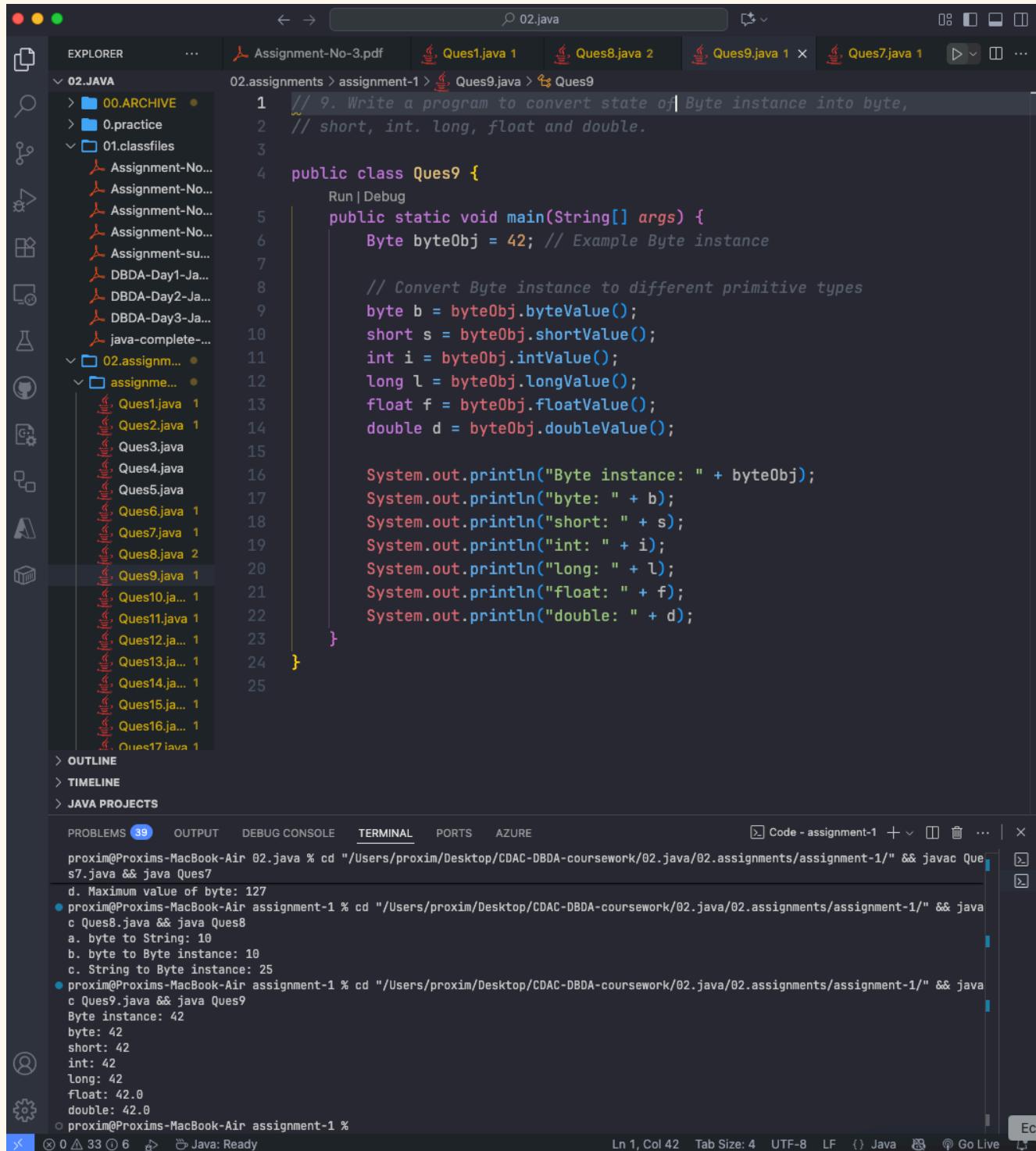
```
1 // 8. Write a program to convert:
2 // a. byte value into String
3 // b. byte value into Byte instance.
4 // c. String instance into Byte instance.
5
6 public class Ques8 {
7     public static void main(String[] args) {
8         // a. byte value into String
9         byte byteValue = 10;
10        String strFromByte = Byte.toString(byteValue);
11        System.out.println("a. byte to String: " + strFromByte);
12
13        // b. byte value into Byte instance
14        Byte byteInstance = Byte.valueOf(byteValue);
15        System.out.println("b. byte to Byte instance: " + byteInstance);
16
17        // c. String instance into Byte instance
18        String strValue = "25";
19        Byte byteFromString = Byte.valueOf(strValue);
20        System.out.println("c. String to Byte instance: " + byteFromString);
21    }
22
23 }
```

The code editor has tabs for Ques1.java 1, Ques8.java 2 (which is the current file), Ques7.java 1, and Ques2.java 1. The terminal tab shows the following command-line session:

```
cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && javac Ques7.java && java Ques7
a. Number of bits in byte: 8
b. Number of bytes in byte: 1
c. Minimum value of byte: -128
d. Maximum value of byte: 127
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && javac Ques8.java && java Ques8
a. byte to String: 10
b. byte to Byte instance: 10
c. String to Byte instance: 25
proxim@Proxims-MacBook-Air assignment-1 %
```

The bottom status bar shows the file is Java Ready, with line 5, column 1, and other standard status indicators.

## Ques 9

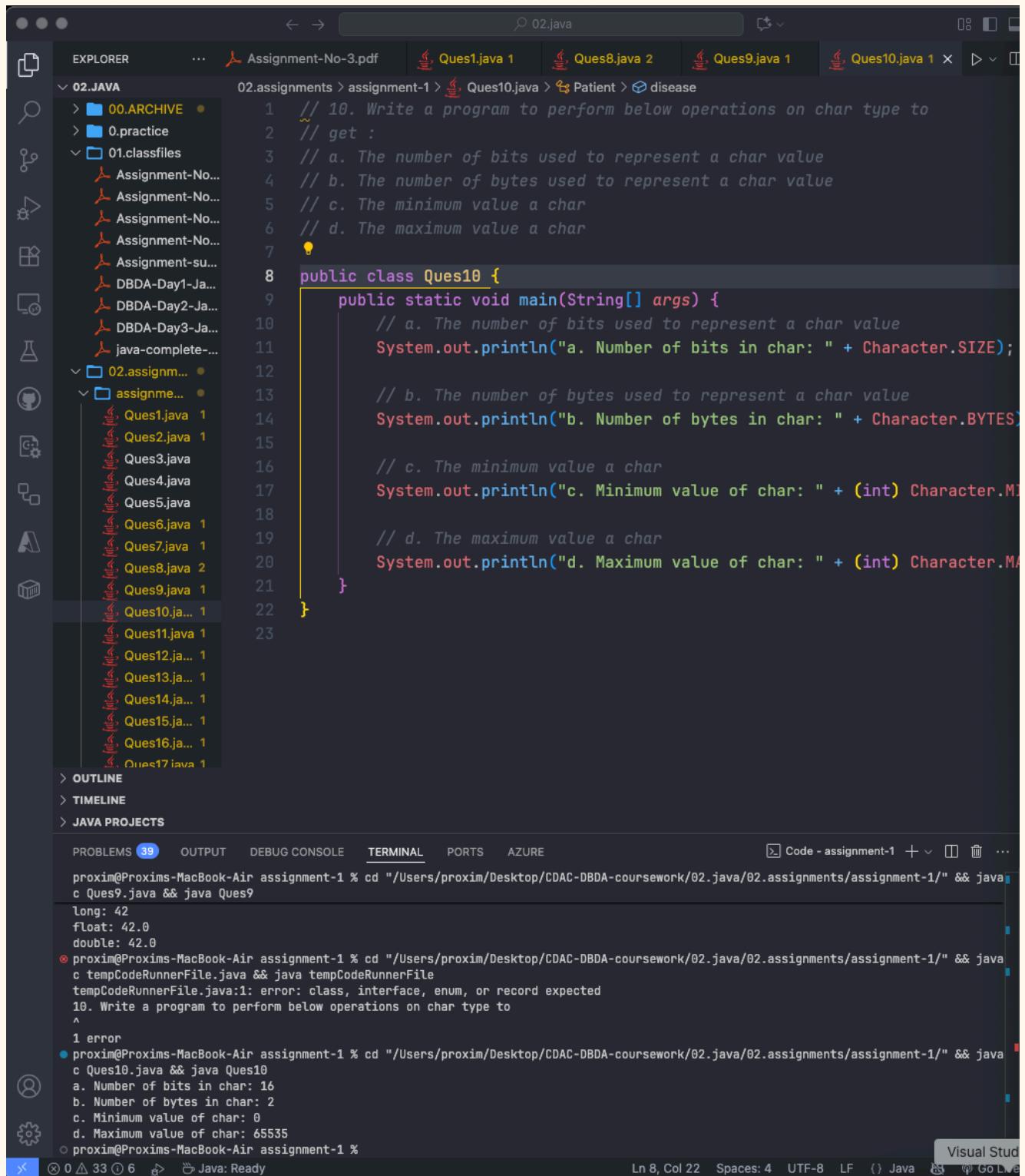


The screenshot shows a Java code editor interface with a dark theme. The code editor displays a Java file named `Ques9.java` with the following content:

```
1 // 9. Write a program to convert state of Byte instance into byte,
2 // short, int, long, float and double.
3
4 public class Ques9 {
5     public static void main(String[] args) {
6         Byte byteObj = 42; // Example Byte instance
7
8         // Convert Byte instance to different primitive types
9         byte b = byteObj.byteValue();
10        short s = byteObj.shortValue();
11        int i = byteObj.intValue();
12        long l = byteObj.longValue();
13        float f = byteObj.floatValue();
14        double d = byteObj.doubleValue();
15
16        System.out.println("Byte instance: " + byteObj);
17        System.out.println("byte: " + b);
18        System.out.println("short: " + s);
19        System.out.println("int: " + i);
20        System.out.println("long: " + l);
21        System.out.println("float: " + f);
22        System.out.println("double: " + d);
23    }
24
25 }
```

The code is a simple program that takes a `Byte` object with value 42 and prints its primitive type equivalents. The code editor's interface includes a sidebar with project navigation, a bottom navigation bar with tabs like PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and AZURE, and a status bar at the bottom.

## Ques 10.



02.java

Assignment-No-3.pdf | Ques1.java 1 | Ques8.java 2 | Ques9.java 1 | Ques10.java 1

02.JAVA

00.ARCHIVE

0.practice

01.classfiles

Assignment-No...

Assignment-No...

Assignment-No...

Assignment-No...

Assignment-No...

Assignment-su...

DBDA-Day1-Ja...

DBDA-Day2-Ja...

DBDA-Day3-Ja...

java-complete...

02.assignm...

assignm...

Ques1.java 1

Ques2.java 1

Ques3.java

Ques4.java

Ques5.java

Ques6.java 1

Ques7.java 1

Ques8.java 2

Ques9.java 1

Ques10.java 1

Ques11.java 1

Ques12.java 1

Ques13.java 1

Ques14.java 1

Ques15.java 1

Ques16.java 1

Ques17.java 1

```
1 // 10. Write a program to perform below operations on char type to
2 // get :
3 // a. The number of bits used to represent a char value
4 // b. The number of bytes used to represent a char value
5 // c. The minimum value a char
6 // d. The maximum value a char
7
8 public class Ques10 {
9     public static void main(String[] args) {
10         // a. The number of bits used to represent a char value
11         System.out.println("a. Number of bits in char: " + Character.SIZE);
12
13         // b. The number of bytes used to represent a char value
14         System.out.println("b. Number of bytes in char: " + Character.BYTES);
15
16         // c. The minimum value a char
17         System.out.println("c. Minimum value of char: " + (int) Character.MIN_VALUE);
18
19         // d. The maximum value a char
20         System.out.println("d. Maximum value of char: " + (int) Character.MAX_VALUE);
21     }
22 }
23
```

OUTLINE

TIMELINE

JAVA PROJECTS

PROBLEMS 39 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java c Ques9.java && java Ques9

long: 42

float: 42.0

double: 42.0

proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java c tempCodeRunnerFile.java && java tempCodeRunnerFile

tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected

10. Write a program to perform below operations on char type to

1 error

proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java c Ques10.java && java Ques10

a. Number of bits in char: 16

b. Number of bytes in char: 2

c. Minimum value of char: 0

d. Maximum value of char: 65535

proxim@Proxims-MacBook-Air assignment-1 %

Visual Studio

Ln 8, Col 22 Spaces: 4 UTF-8 LF {} Java Go Live

## Ques 11.



The screenshot shows a Java code editor with the following details:

- File Explorer:** Shows a project structure under "02.JAVA" with various Java files (Ques1.java, Ques8.java, Ques9.java, Ques10.java, Ques11.java) and a "00.ARCHIVE" folder.
- Code Editor:** The "Ques11.java" file is open, containing the following code:

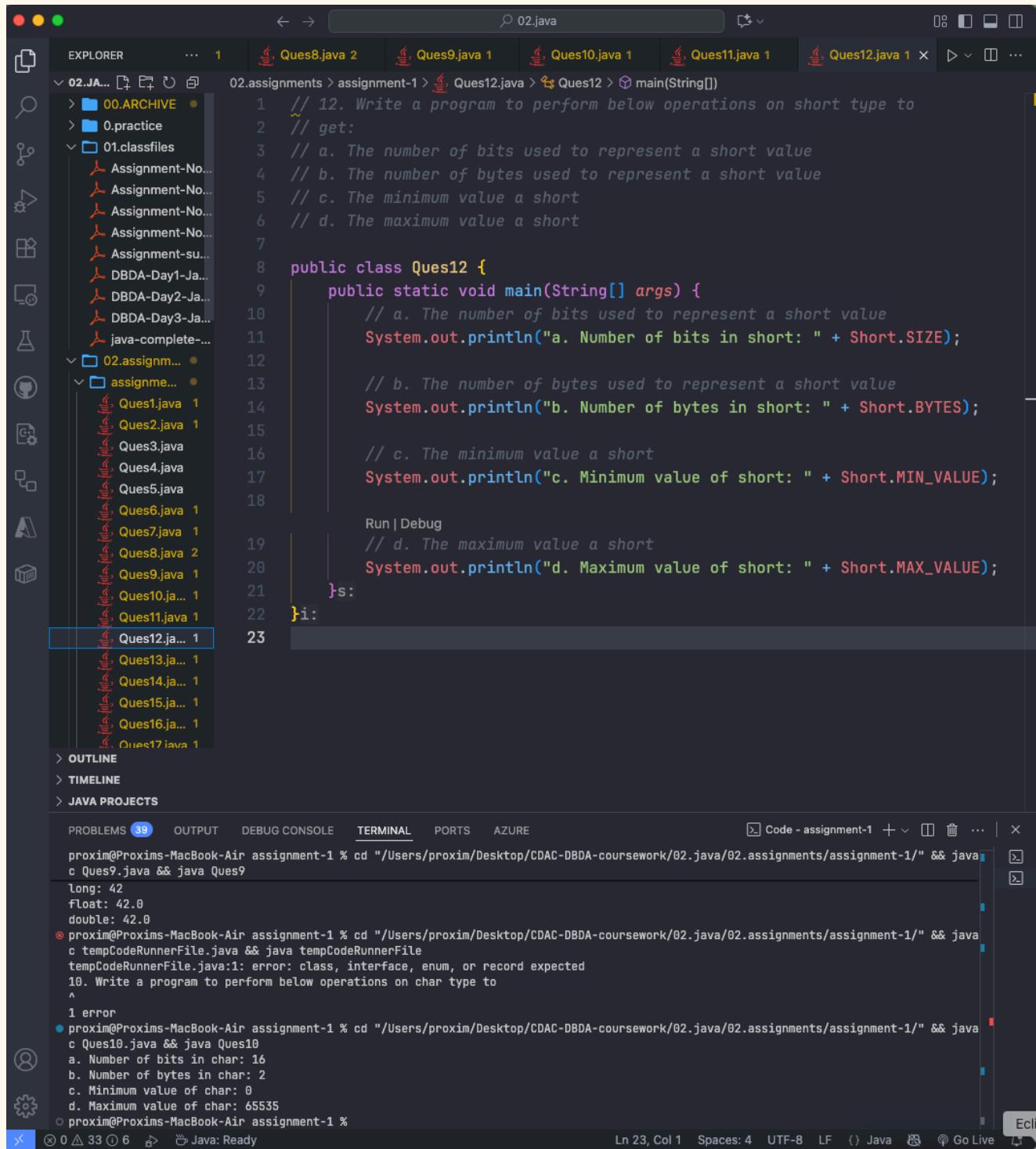
```
1 // 11. Accept character from command line and perform below
2 // operations. Here you can use charAt() method to extract
3 // character:
4 // a. Check whether entered character is letter or digit. If it
5 // is digit then print its values as well as code point.
6 // b. If it is character then check whether it is in lowercase?
7 // If it is in lowercase then convert it into upper case and
8 // print it well as its code point. If it is in uppercase
9 // then convert it into lower case and print it well as its
10 // code point.
11
12 public class Ques11 {
13     public static void main(String[] args) {
14         if (args.length == 0 || args[0].length() == 0) {
15             System.out.println("Please provide a character as a command line arg");
16             return;
17         }
18         char ch = args[0].charAt(0);
19
20         Run | Debug
21         if (Character.isDigit(ch)) {
22             System.out.println("Entered character is a digit.");
23             System.out.println("Value: " + ch);
24             System.out.println("Code point: " + (int) ch);
25         } else if (Character.isLetter(ch)) {
26             System.out.println("Entered character is a letter.");
27             if (Character.isLowerCase(ch)) {
28                 char upper = Character.toUpperCase(ch);
29                 System.out.println("Lowercase detected. Converted to uppercase:");
30                 System.out.println("Code point: " + (int) upper);
31             } else if (Character.isUpperCase(ch)) {
```

- Terminal:** Shows the command line output of running the program:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques9.java && java Ques9
long: 42
float: 42.0
double: 42.0
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
10. Write a program to perform below operations on char type to
^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques10
a. Number of bits in char: 16
b. Number of bytes in char: 2
c. Minimum value of char: 0
d. Maximum value of char: 65535
proxim@Proxims-MacBook-Air assignment-1 %
```

- Status Bar:** Shows the current line (Ln 10, Col 15), spaces (Spaces: 4), encoding (UTF-8), line separator (LF), Java mode, and Go Live status.

## Ques 12.



The screenshot shows an IDE interface with the following details:

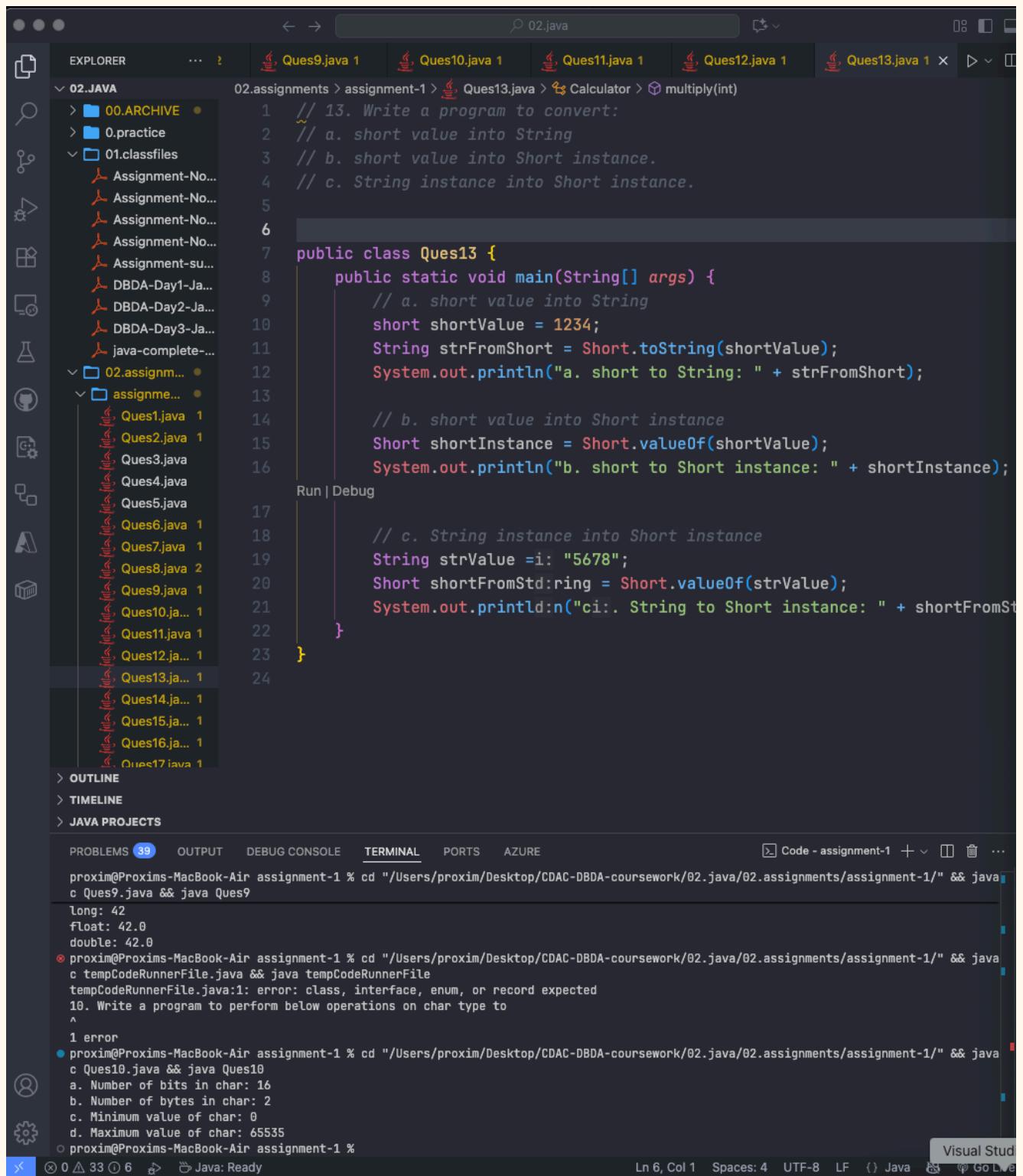
- File Bar:** Shows tabs for Ques8.java 2, Ques9.java 1, Ques10.java 1, Ques11.java 1, Ques12.java 1 (highlighted in yellow), and Ques13.java 1.
- Explorer:** Shows a tree view of files and folders under '02.JAVA...'. The '01.classfiles' folder contains sub-folders for Assignment-No... and Assignment-su... with various Java files (Ques1.java 1, Ques2.java 1, etc.).
- Code Editor:** The main editor window contains Java code for 'Ques12.java'. The code is as follows:

```
1 // 12. Write a program to perform below operations on short type to
2 // get:
3 // a. The number of bits used to represent a short value
4 // b. The number of bytes used to represent a short value
5 // c. The minimum value a short
6 // d. The maximum value a short
7
8 public class Ques12 {
9     public static void main(String[] args) {
10         // a. The number of bits used to represent a short value
11         System.out.println("a. Number of bits in short: " + Short.SIZE);
12
13         // b. The number of bytes used to represent a short value
14         System.out.println("b. Number of bytes in short: " + Short.BYTES);
15
16         // c. The minimum value a short
17         System.out.println("c. Minimum value of short: " + Short.MIN_VALUE);
18
19         Run | Debug
20         // d. The maximum value a short
21         System.out.println("d. Maximum value of short: " + Short.MAX_VALUE);
22     }
23 }
```

- Terminal:** The terminal window shows the following command and output:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques9
long: 42
float: 42.0
double: 42.0
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
10. Write a program to perform below operations on char type to
^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques10
a. Number of bits in char: 16
b. Number of bytes in char: 2
c. Minimum value of char: 0
d. Maximum value of char: 65535
proxim@Proxims-MacBook-Air assignment-1 %
```
- Bottom Status Bar:** Shows 'Ln 23, Col 1' and other status indicators.

### Ques 13.



The screenshot shows a Visual Studio Code interface with the following details:

- Explorer View:** Shows a tree structure for the "02.JAVA" project. The "02.assignments" folder contains sub-folders for "00.ARCHIVE", "0.practice", and "01.classfiles". The "01.classfiles" folder contains several assignment files: Assignment-No..., Assignment-No..., Assignment-No..., Assignment-No..., Assignment-No..., Assignment-su..., DBDA-Day1-Ja..., DBDA-Day2-Ja..., DBDA-Day3-Ja..., and java-complete-.... The "02.assignments" folder also contains sub-folders for "02.assignments" and "assignme...", which in turn contain files like Ques1.java, Ques2.java, Ques3.java, Ques4.java, Ques5.java, Ques6.java, Ques7.java, Ques8.java, Ques9.java, Ques10.java, Ques11.java, Ques12.java, Ques13.java, Ques14.java, Ques15.java, Ques16.java, and Ques17.java.
- Code Editor:** The main editor window displays the Java code for "Ques13.java". The code is as follows:

```
1 // 13. Write a program to convert:
2 // a. short value into String
3 // b. short value into Short instance.
4 // c. String instance into Short instance.
5
6 public class Ques13 {
7     public static void main(String[] args) {
8         // a. short value into String
9         short shortValue = 1234;
10        String strFromShort = Short.toString(shortValue);
11        System.out.println("a. short to String: " + strFromShort);
12
13        // b. short value into Short instance
14        Short shortInstance = Short.valueOf(shortValue);
15        System.out.println("b. short to Short instance: " + shortInstance);
16
17        // c. String instance into Short instance
18        String strValue = "5678";
19        Short shortFromString = Short.valueOf(strValue);
20        System.out.println("c. String to Short instance: " + shortFromString);
21    }
22
23
24 }
```

- Terminal:** The terminal shows the following command and output:

```
proxim@Proxim-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques9.java && java Ques9
long: 42
float: 42.0
double: 42.0
proxim@Proxim-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
10. Write a program to perform below operations on char type to
^
1 error
proxim@Proxim-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques10.java && java Ques10
a. Number of bits in char: 16
b. Number of bytes in char: 2
c. Minimum value of char: 0
d. Maximum value of char: 65535
proxim@Proxim-MacBook-Air assignment-1 %
```
- Status Bar:** The status bar at the bottom shows "Ln 6, Col 1" and "Java: Ready".

## Ques 14.:

02.java

EXPLORER    ...

02.JAVA

02.assignm... ●

  └ assignme... ●

    Ques4.java

    Ques5.java

    Ques6.java 1

    Ques7.java 1

    Ques8.java 2

    Ques9.java 1

    Ques10.java... 1

    Ques11.java 1

    Ques12.java... 1

    Ques13.java... 1

    Ques14.java... 1

    Ques15.java... 1

    Ques16.java... 1

    Ques17.java 1

    Ques18.java... 1

    Ques19.java... 1

    Ques20.java... 1

    Ques21.java... 1

    Ques22.java... 1

    Ques23.java... 1

    Ques24.java... 1

    Ques25.java... 1

    Ques26.java... 1

    Ques27.java... 1

    Ques28.java... 1

    Ques29.java... 1

    Ques30.java... 1

    Ques31.java... 1

    Ques32.java... 1

Ques11.java 1    Ques12.java 1    Ques13.java 1    Ques15.java 1    Ques14.java 1 X

02.assignm... > assignment-1 > Ques14.java > Ques14

1 // 14. Write a program to convert state of Short instance into byte,  
2 // short, int, long, float and double.

3

4 public class Ques14 {

5     public static void main(String[] args) {

6         Short shortObj = 1234; // Example Short instance

7

8         byte b = shortObj.byteValue();

9         short s = shortObj.shortValue();

10        int i = shortObj.intValue();

11        long l = shortObj.longValue();

12        float f = shortObj.floatValue();

13        double d = shortObj.doubleValue();

14

15        Run | Debug

16        System.out.println("Short instance: " + shortObj);

17        System.out.println("byte: " + b);

18        System.out.println("short: " + s);

19        System.out.println("int: " + i);

20        System.out.println("long: " + l);

21        System.out.println("float: " + f);

22        System.out.println("double: " + d);

23     }

24 }

Run | Debug

System.out.println("Short instance: " + shortObj);

System.out.println("byte: " + b);

System.out.println("short: " + s);

System.out.println("int: " + i);

System.out.println("long: " + l);

System.out.println("float: " + f);

System.out.println("double: " + d);

OUTLINE

TIMELINE

JAVA PROJECTS

PROBLEMS 39    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    AZURE

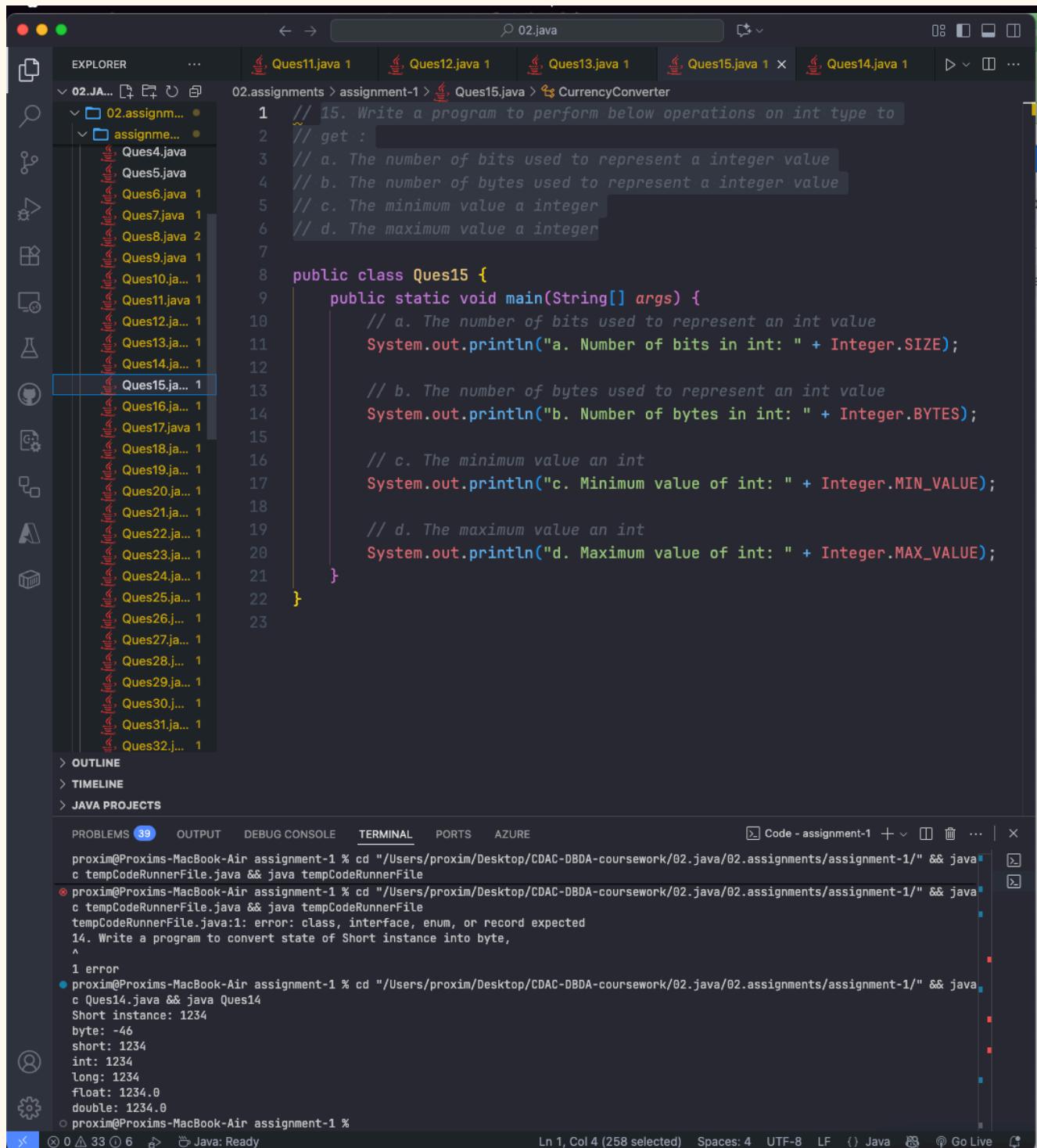
Code - assignment-1

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
14. Write a program to convert state of Short instance into byte,
 ^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques14
Short instance: 1234
byte: -46
short: 1234
int: 1234
long: 1234
float: 1234.0
double: 1234.0
proxim@Proxims-MacBook-Air assignment-1 %
```

0 33 6 Java Ready

Line 19, Col 42    Spaces: 4    UTF-8    LF    ⌂.java    ⌂.golve    ⌂.Golve

## Ques 15.



The screenshot shows a Java code editor with the following details:

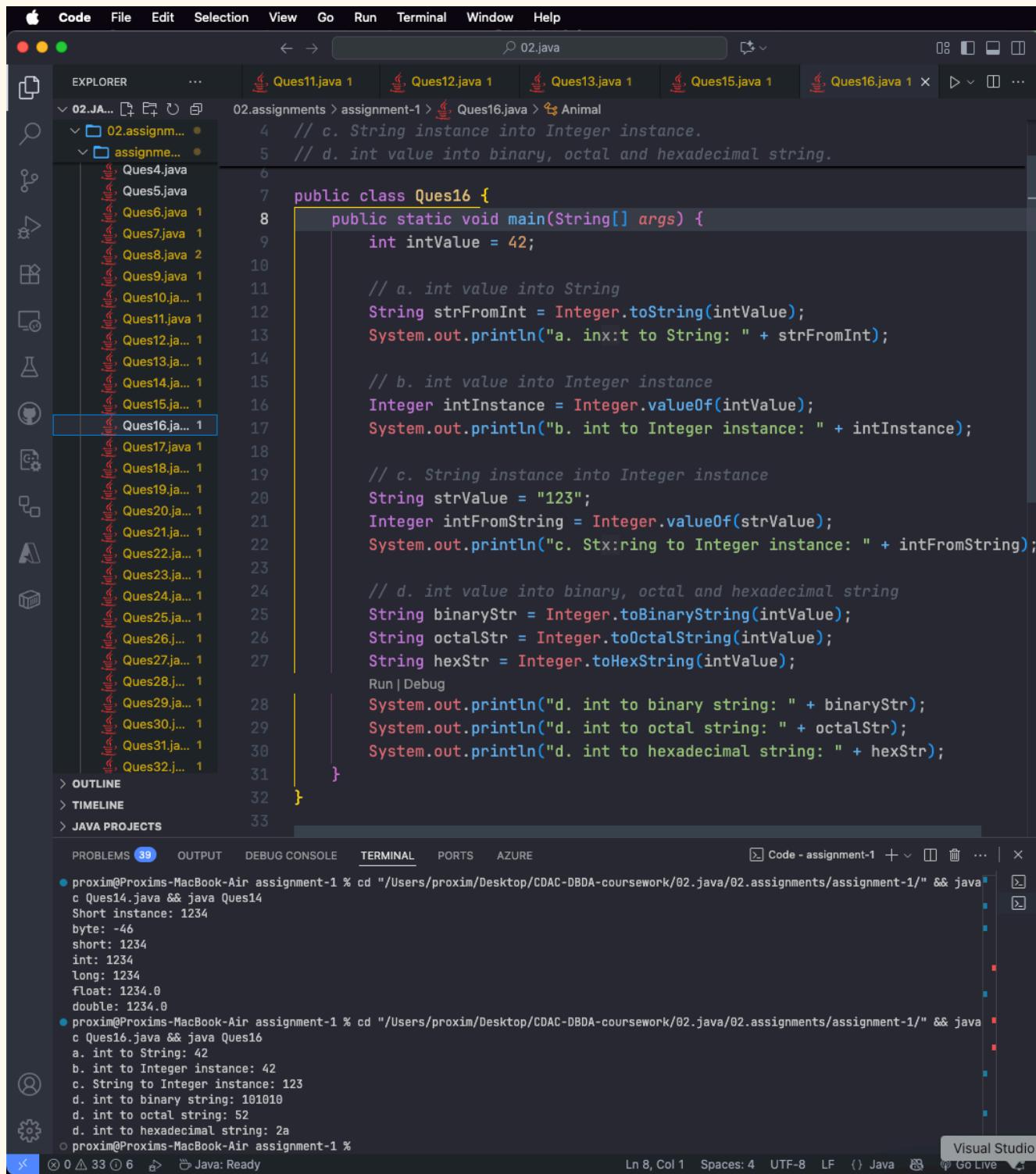
- File Explorer:** Shows a tree structure of Java files under the path `02.assignments > assignment-1 > Ques15.java`. The file `Ques15.java` is selected.
- Code Editor:** The content of `Ques15.java` is displayed. It contains a class `Ques15` with a `main` method. The code prints the number of bits, bytes, minimum, and maximum values for an `int` type.
- Terminal:** The terminal shows the following command and output:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
```

Output:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
14. Write a program to convert state of Short instance into byte,
^
1 error
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques14.java && java Ques14
Short instance: 1234
byte: -46
short: 1234
int: 1234
long: 1234
float: 1234.0
double: 1234.0
○ proxim@Proxims-MacBook-Air assignment-1 %
```
- Bottom Status:** Shows the terminal command, line 1, column 4 (258 selected), spaces: 4, UTF-8, LF, Java, Go Live, and a refresh icon.

## Ques 16.



The screenshot shows the Visual Studio Code interface with the following details:

- Code Editor:** The main pane displays the Java code for `Ques16.java`. The code demonstrates various Java primitive type conversions and string representations.
- Explorer:** The left sidebar shows a file tree for the `02.JA...` project, including files like `Ques11.java`, `Ques12.java`, `Ques13.java`, `Ques15.java`, and `Ques16.java`.
- Terminal:** The bottom terminal pane shows the output of running the code, displaying the conversion of the integer value 42 into various string representations (byte, short, int, long, float, double) and then back into integer instances.

```
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques14
c Ques14.java && java Ques14
Short instance: 1234
byte: -46
short: 1234
int: 1234
long: 1234
float: 1234.0
double: 1234.0
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques16
a. int to String: 42
b. int to Integer instance: 42
c. String to Integer instance: 123
d. int to binary string: 101010
d. int to octal string: 52
d. int to hexadecimal string: 2a
○ proxim@Proxims-MacBook-Air assignment-1 %
```

- Bottom Status Bar:** The status bar shows the current line (Ln 8, Col 1), character count (Spaces: 4), encoding (UTF-8), file type (Java), and other status indicators.

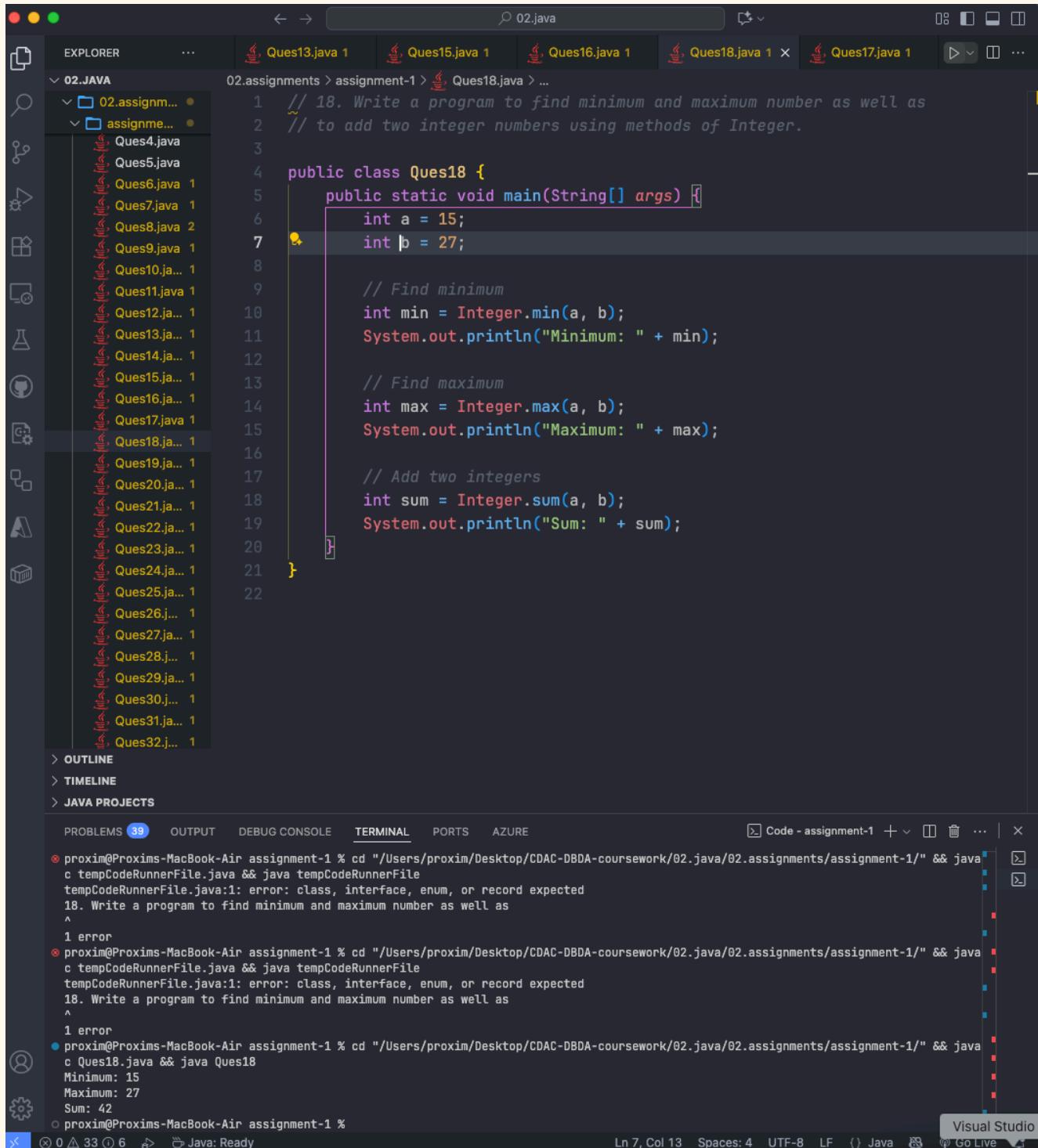
## Ques 17.

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a project structure under "02.JA...". The "02.assignm..." folder contains "Ques17.java 1". Other files in the folder include Ques4.java, Ques5.java, Ques6.java 1, Ques7.java 1, Ques8.java 2, Ques9.java 1, Ques10.java... 1, Ques11.java 1, Ques12.java... 1, Ques13.java... 1, Ques14.java... 1, Ques15.java... 1, Ques16.java... 1, Ques17.java 1, Ques18.java... 1, Ques19.java... 1, Ques20.java... 1, Ques21.java... 1, Ques22.java... 1, Ques23.java... 1, Ques24.java... 1, Ques25.java... 1, Ques26.java... 1, Ques27.java... 1, Ques28.java... 1, Ques29.java... 1, Ques30.java... 1, Ques31.java... 1, and Ques32.java... 1.
- Code Editor:** The main editor window displays the Java code for Ques 17. The code converts an Integer instance to byte, short, int, long, float, and double and prints them to the console.
- Terminal:** The terminal window shows the execution of the program. It outputs:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques17
Short instance: 1234
byte: -46
short: 1234
int: 1234
long: 1234
float: 1234.0
double: 1234.0
```
- Bottom Status Bar:** The status bar shows "Ln 1, Col 48" and "Java: Ready".

## Ques 18.



The screenshot shows a Visual Studio Code interface with the following details:

- Explorer:** Shows a file tree for a project named "02.JAVA" under "02.assignments > assignment-1". The file "Ques18.java" is selected.
- Code Editor:** Displays the Java code for Ques 18. The code uses the Integer class to find the minimum and maximum of two integers (a and b) and to add them.
- Terminal:** Shows the command-line output of running the code. It includes compilation errors for a tempCodeRunnerFile.java and successful execution of Ques18.java.
- Status Bar:** Shows the current line (Ln 7, Col 13), spaces (4), encoding (UTF-8), line feed (LF), Java, and Go Live status.

```
// 18. Write a program to find minimum and maximum number as well as
// to add two integer numbers using methods of Integer.

public class Ques18 {
    public static void main(String[] args) {
        int a = 15;
        int b = 27;

        // Find minimum
        int min = Integer.min(a, b);
        System.out.println("Minimum: " + min);

        // Find maximum
        int max = Integer.max(a, b);
        System.out.println("Maximum: " + max);

        // Add two integers
        int sum = Integer.sum(a, b);
        System.out.println("Sum: " + sum);
    }
}
```

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
18. Write a program to find minimum and maximum number as well as
^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
18. Write a program to find minimum and maximum number as well as
^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques18.java && java Ques18
Minimum: 15
Maximum: 27
Sum: 42
proxim@Proxims-MacBook-Air assignment-1 %
```

## Ques 19.



02.assignments > assignment-1 > Ques19.java > ...

```
1 // 19. Write a program to perform below operations on long type to
2 // get:
3 // a. The number of bits used to represent a long value
4 // b. The number of bytes used to represent a long value
5 // c. The minimum value a long
6 // d. The maximum value a long
7
8 public class Ques19 {
9     public static void main(String[] args) {
10         // a. The number of bits used to represent a long value
11         System.out.println("a. Number of bits in long: " + Long.SIZE);
12
13         // b. The number of bytes used to represent a long value
14         System.out.println("b. Number of bytes in long: " + Long.BYTES);
15
16         // c. The minimum value a long
17         System.out.println("c. Minimum value of long: " + Long.MIN_VALUE);
18
19         // d. The maximum value a long
20         System.out.println("d. Maximum value of long: " + Long.MAX_VALUE);
21     }
22 }
23
```

Run | Debug

PROBLEMS 39 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
18. Write a program to find minimum and maximum number as well as
 ^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
18. Write a program to find minimum and maximum number as well as
 ^
1 error
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques18
Ques18.java & java Ques18
Minimum: 15
Maximum: 27
Sum: 42
○ proxim@Proxims-MacBook-Air assignment-1 %

```

Ln 8, Col 22 Spaces: 4 UTF-8 LF ( Java Go Live

Ques 20.

Ques15.java 1 | Ques16.java 1 | Ques18.java 1 | Ques20.java 1 X | Ques19.java 1

02.JAVA

02.assignm... | 02.assignm...

Ques4.java | Ques5.java | Ques6.java 1 | Ques7.java 1 | Ques8.java 2 | Ques9.java 1 | Ques10.java 1 | Ques11.java 1 | Ques12.java 1 | Ques13.java 1 | Ques14.java 1 | Ques15.java 1 | Ques16.java 1 | Ques17.java 1 | Ques18.java 1 | Ques19.java 1 | Ques20.java 1 | Ques21.java 1 | Ques22.java 1 | Ques23.java 1 | Ques24.java 1 | Ques25.java 1 | Ques26.java 1 | Ques27.java 1 | Ques28.java 1 | Ques29.java 1 | Ques30.java 1 | Ques31.java 1 | Ques32.java 1

02.assignments > assignment-1 > Ques20.java > Ques20 > main(String[])

```
1 // 20. Write a program to convert:
2 // a. long value into String
3 // b. long value into Long instance.
4 // c. String instance into Long instance.
5 // d. long value into binary, octal and hexadecimal string.
6
7 public class Ques20 {
8     public static void main(String[] args) {
9         long longValue = 123456789L;
10
11     // a. long value into String
12     String strFromLong = Long.toString(longValue);
13     System.out.println("a. long to String: " + strFromLong);
14
15     // b. long value into Long instance
16     Long longInstance = Long.valueOf(longValue);
17     System.out.println("b. long to Long instance: " + longInstance);
18
19     // c. String instance into Long instance
20     String strValue = "987654321";
21     Long longFromString = Long.valueOf(strValue);
22     System.out.println("c. String to Long instance: " + longFromString);
23
24     // d. long value into binary, octal and hexadecimal string
25     String binaryStr = Long.toBinaryString(longValue);
26     String octalStr = Long.toOctalString(longValue);
27     String hexStr = Long.toHexString(longValue);
28     Run | Debug
29     System.out.println("d. long to binary string: " + binaryStr);
30     System.out.println("d. long to octal string: " + octalStr);
31     System.out.println("d. long to hexadecimal string: " + hexStr);
32 }
```

OUTLINE

TIMELINE

JAVA PROJECTS

PROBLEMS 39

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

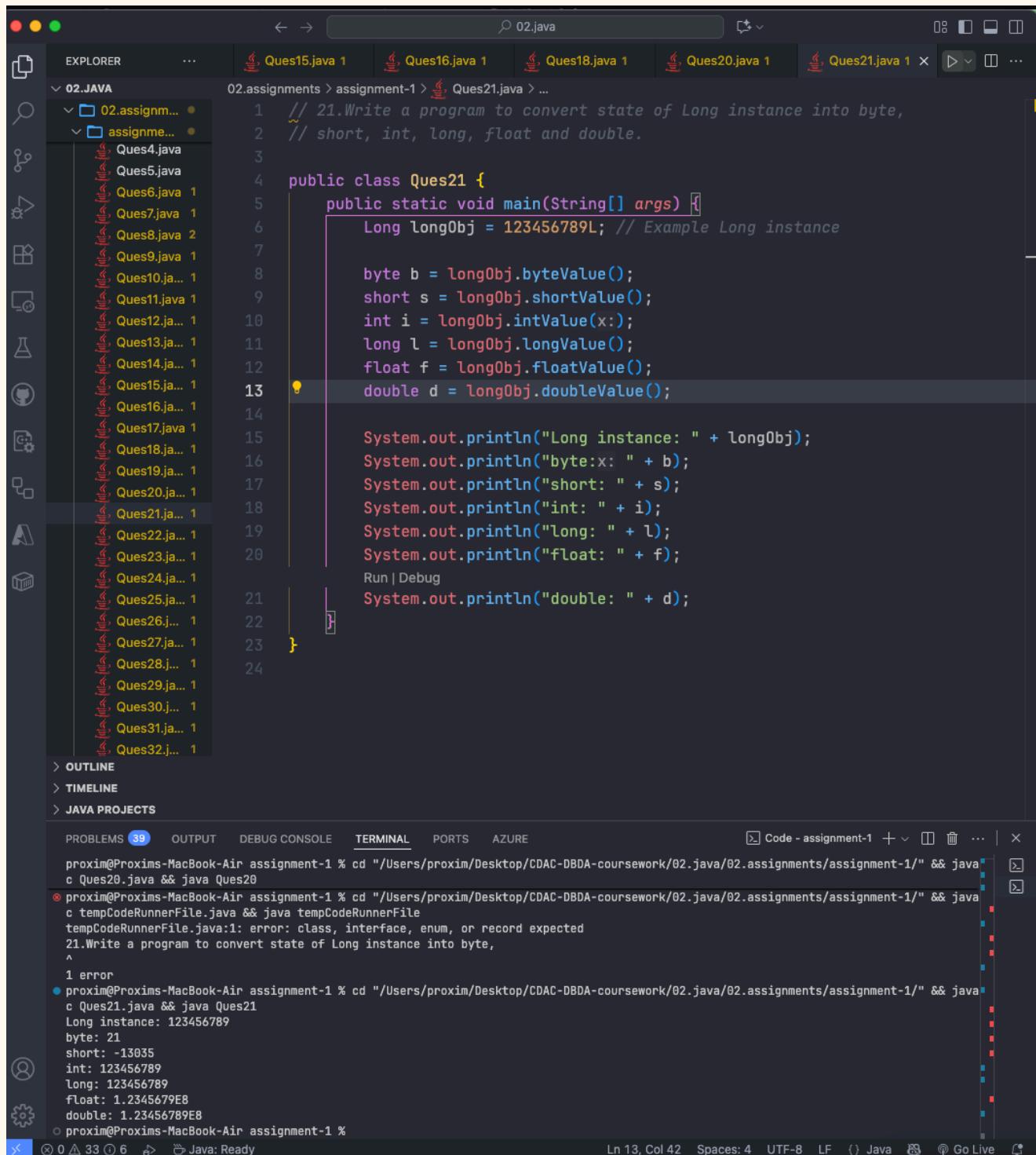
AZURE

Code - assignment-1

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java c tempCodeRunnerFile.java && java tempCodeRunnerFile
^
1 error
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java c Ques18.java && java Ques18
Minimum: 15
Maximum: 27
Sum: 42
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java c Ques20.java && java Ques20
a. long to String: 123456789
b. long to Long instance: 123456789
c. String to Long instance: 987654321
d. long to binary string: 11010110111100110100010101
d. long to octal string: 726746425
d. long to hexadecimal string: 75bcd15
○ proxim@Proxims-MacBook-Air assignment-1 %
```

Ln 33, Col 1 | Spaces: 4 | UTF-8 | LF | {} Java | Go Live

## Ques 21.

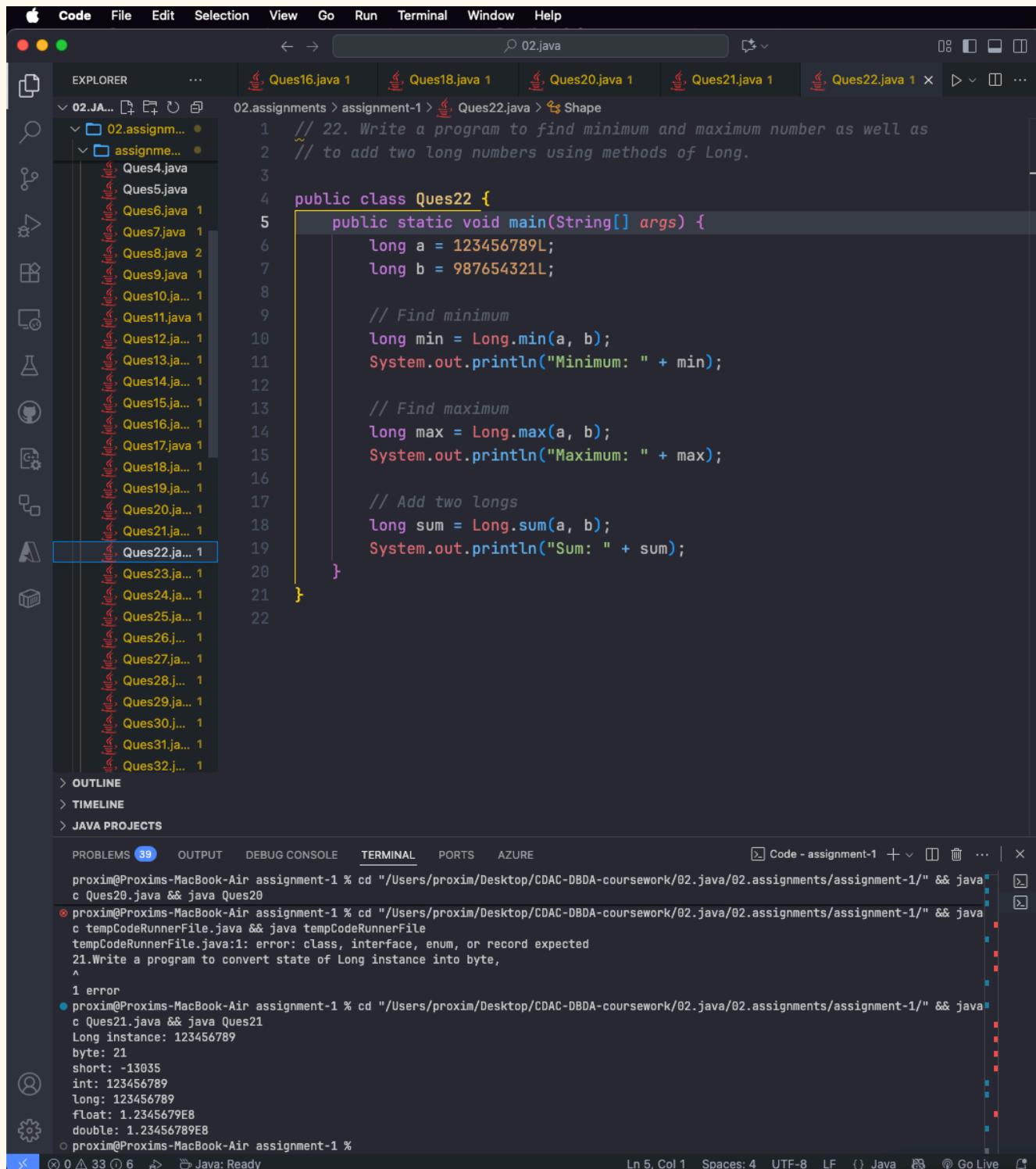


The screenshot shows a Java code editor with the following code:

```
1 // 21. Write a program to convert state of Long instance into byte,
2 // short, int, long, float and double.
3
4 public class Ques21 {
5     public static void main(String[] args) {
6         Long longObj = 123456789L; // Example Long instance
7
8         byte b = longObj.byteValue();
9         short s = longObj.shortValue();
10        int i = longObj.intValue();
11        long l = longObj.longValue();
12        float f = longObj.floatValue();
13        double d = longObj.doubleValue();
14
15        System.out.println("Long instance: " + longObj);
16        System.out.println("byte: " + b);
17        System.out.println("short: " + s);
18        System.out.println("int: " + i);
19        System.out.println("long: " + l);
20        System.out.println("float: " + f);
21        Run | Debug
22        System.out.println("double: " + d);
23    }
24}
```

The code is part of a project named "02.assignments" under "assignment-1". The "Ques21.java" file is selected in the Explorer view. The code prints the converted values of a Long object to the console. The code editor interface includes a terminal window showing the execution of the program, which outputs the converted values.

## Ques 22.



The screenshot shows a Java code editor with the following details:

- File Explorer:** Shows a project structure under "02.JA...". The "02.assignm..." folder contains numerous Java files (Ques4.java, Ques5.java, etc.) and a "assignment-1" folder containing "Ques22.java".
- Code Editor:** The "Ques22.java" file is open. The code is as follows:

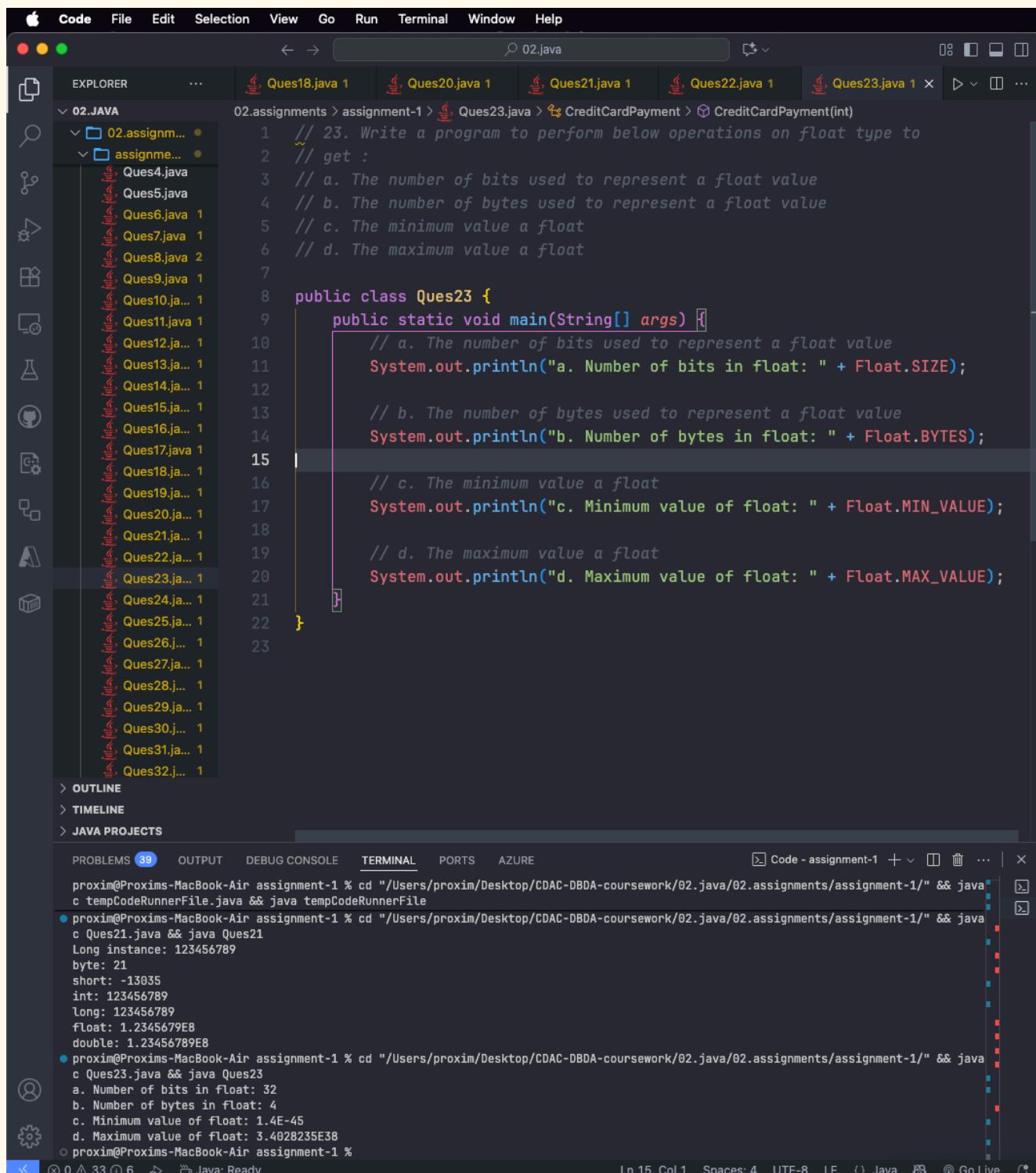
```
1 // 22. Write a program to find minimum and maximum number as well as
2 // to add two long numbers using methods of Long.
3
4 public class Ques22 {
5     public static void main(String[] args) {
6         long a = 123456789L;
7         long b = 987654321L;
8
9         // Find minimum
10        long min = Long.min(a, b);
11        System.out.println("Minimum: " + min);
12
13        // Find maximum
14        long max = Long.max(a, b);
15        System.out.println("Maximum: " + max);
16
17        // Add two longs
18        long sum = Long.sum(a, b);
19        System.out.println("Sum: " + sum);
20    }
21
22 }
```

- Terminal:** Shows the following command-line session:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques20.java && java Ques20
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
21. Write a program to convert state of Long instance into byte,
^
1 error
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques21.java && java Ques21
Long instance: 123456789
byte: 21
short: -13035
int: 123456789
long: 123456789
float: 1.2345679E8
double: 1.23456789E8
proxim@Proxims-MacBook-Air assignment-1 %
```

- Status Bar:** Shows "Java: Ready" and other terminal details.

## Ques 23.



The screenshot shows a Java code editor interface with the following details:

- File Menu:** Code, File, Edit, Selection, View, Go, Run, Terminal, Window, Help.
- Toolbar:** Standard Mac OS X style with red, yellow, and green buttons.
- Code Editor:** The active tab is "02.java". The code is as follows:

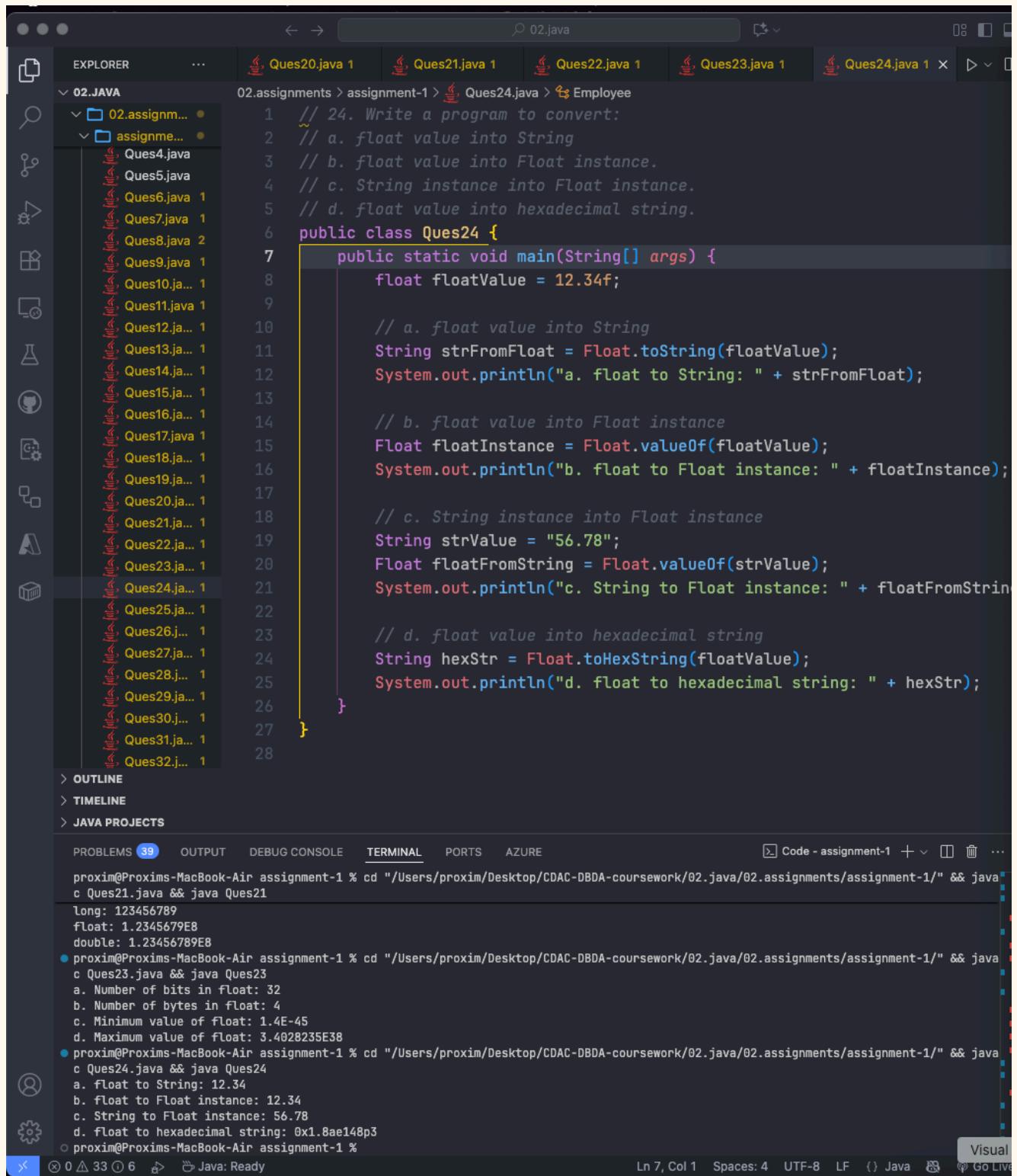
```
1 // 23. Write a program to perform below operations on float type to
2 // get :
3 // a. The number of bits used to represent a float value
4 // b. The number of bytes used to represent a float value
5 // c. The minimum value a float
6 // d. The maximum value a float
7
8 public class Ques23 {
9     public static void main(String[] args) {
10         // a. The number of bits used to represent a float value
11         System.out.println("a. Number of bits in float: " + Float.SIZE);
12
13         // b. The number of bytes used to represent a float value
14         System.out.println("b. Number of bytes in float: " + Float.BYTES);
15
16         // c. The minimum value a float
17         System.out.println("c. Minimum value of float: " + Float.MIN_VALUE);
18
19         // d. The maximum value a float
20         System.out.println("d. Maximum value of float: " + Float.MAX_VALUE);
21     }
22 }
23
```

- Explorer:** Shows a tree view of files under "02.JAVA".
- Terminal:** Shows the output of running the code:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques21.java && java Ques21
Long instance: 123456789
byte: 21
short: -13035
int: 123456789
long: 123456789
float: 1.2345679E8
double: 1.23456789E8
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques23.java && java Ques23
a. Number of bits in float: 32
b. Number of bytes in float: 4
c. Minimum value of float: 1.4E-45
d. Maximum value of float: 3.4028235E38
○ proxim@Proxims-MacBook-Air assignment-1 %
```

- Bottom Status:** Shows the current file is "02.java" and the Java environment is ready.

## Ques 24.



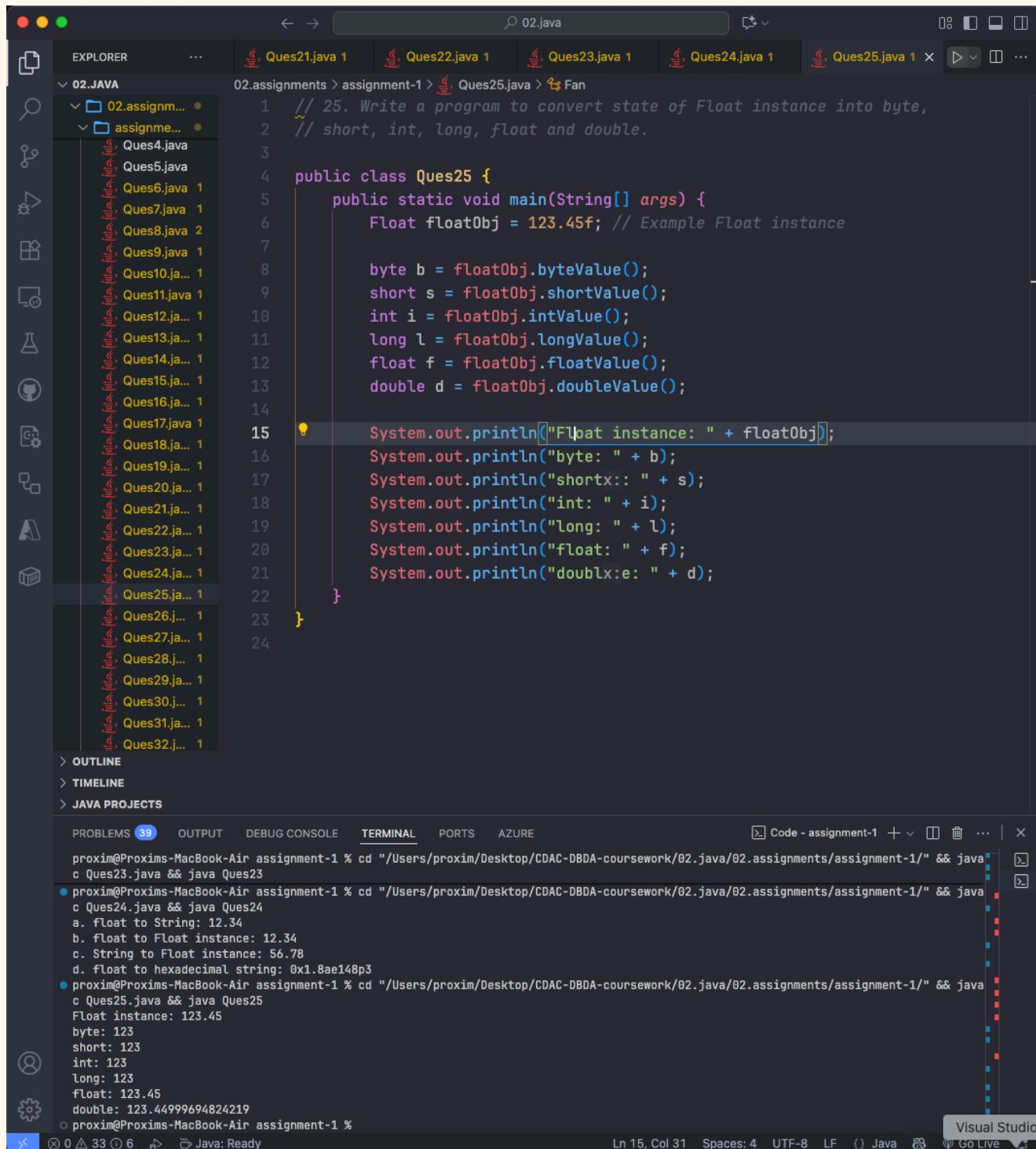
The screenshot shows a Java code editor with the following code:

```
// 24. Write a program to convert:  
// a. float value into String  
// b. float value into Float instance.  
// c. String instance into Float instance.  
// d. float value into hexadecimal string.  
  
public class Ques24 {  
    public static void main(String[] args) {  
        float floatValue = 12.34f;  
  
        // a. float value into String  
        String strFromFloat = Float.toString(floatValue);  
        System.out.println("a. float to String: " + strFromFloat);  
  
        // b. float value into Float instance  
        Float floatInstance = Float.valueOf(floatValue);  
        System.out.println("b. float to Float instance: " + floatInstance);  
  
        // c. String instance into Float instance  
        String strValue = "56.78";  
        Float floatFromString = Float.valueOf(strValue);  
        System.out.println("c. String to Float instance: " + floatFromString);  
  
        // d. float value into hexadecimal string  
        String hexStr = Float.toHexString(floatValue);  
        System.out.println("d. float to hexadecimal string: " + hexStr);  
    }  
}
```

The code is part of a class named `Ques24`. It contains a `main` method that prints four different representations of the float value `12.34f` to the console. The representations are: a. float to String, b. float to Float instance, c. String to Float instance, and d. float to hexadecimal string.

The code editor interface includes an Explorer sidebar on the left showing a list of Java files, a terminal at the bottom showing command-line output, and various status indicators at the bottom right.

## Ques 25.



The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a tree view of Java files in the "02.JAVA" folder, including Ques25.java and many other QuesX.java files.
- Code Editor:** The active file is Ques25.java. The code is as follows:

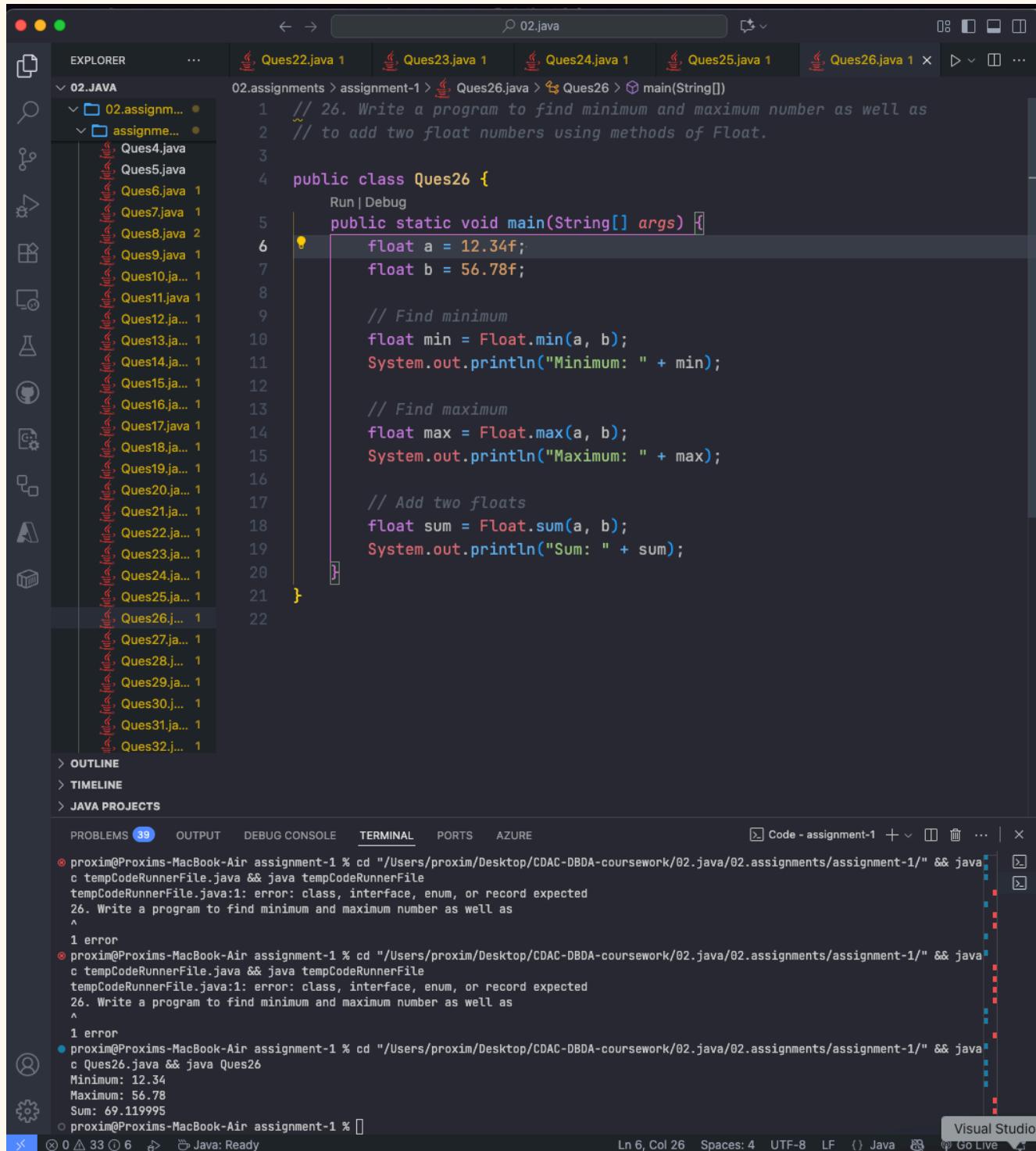
```
1 // 25. Write a program to convert state of Float instance into byte,
2 // short, int, long, float and double.
3
4 public class Ques25 {
5     public static void main(String[] args) {
6         Float floatObj = 123.45f; // Example Float instance
7
8         byte b = floatObj.byteValue();
9         short s = floatObj.shortValue();
10        int i = floatObj.intValue();
11        long l = floatObj.longValue();
12        float f = floatObj.floatValue();
13        double d = floatObj.doubleValue();
14
15        System.out.println("Float instance: " + floatObj);
16        System.out.println("byte: " + b);
17        System.out.println("short: " + s);
18        System.out.println("int: " + i);
19        System.out.println("long: " + l);
20        System.out.println("float: " + f);
21        System.out.println("double: " + d);
22    }
23 }
24
```

- Terminal:** Shows the command line output of running the program:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques23
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques24
a. Float to String: 12.34
b. Float to Float instance: 12.34
c. String to Float instance: 56.78
d. Float to hexadecimal string: 0x1.8ae148p3
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques25
Float instance: 123.45
byte: 123
short: 123
int: 123
long: 123
float: 123.45
double: 123.44999694824219
```

- Status Bar:** Shows the current line (Ln 15, Col 31), spaces (4), encoding (UTF-8), line feed (LF), Java, and Go Live status.

## Ques 26.



The screenshot shows a Java code editor in Visual Studio Code. The code is for a program that finds the minimum and maximum of two floats and adds them. The code uses the `Float` class methods `min`, `max`, and `sum`.

```
// 26. Write a program to find minimum and maximum number as well as
// to add two float numbers using methods of Float.

public class Ques26 {
    public static void main(String[] args) {
        float a = 12.34f;
        float b = 56.78f;

        // Find minimum
        float min = Float.min(a, b);
        System.out.println("Minimum: " + min);

        // Find maximum
        float max = Float.max(a, b);
        System.out.println("Maximum: " + max);

        // Add two floats
        float sum = Float.sum(a, b);
        System.out.println("Sum: " + sum);
    }
}
```

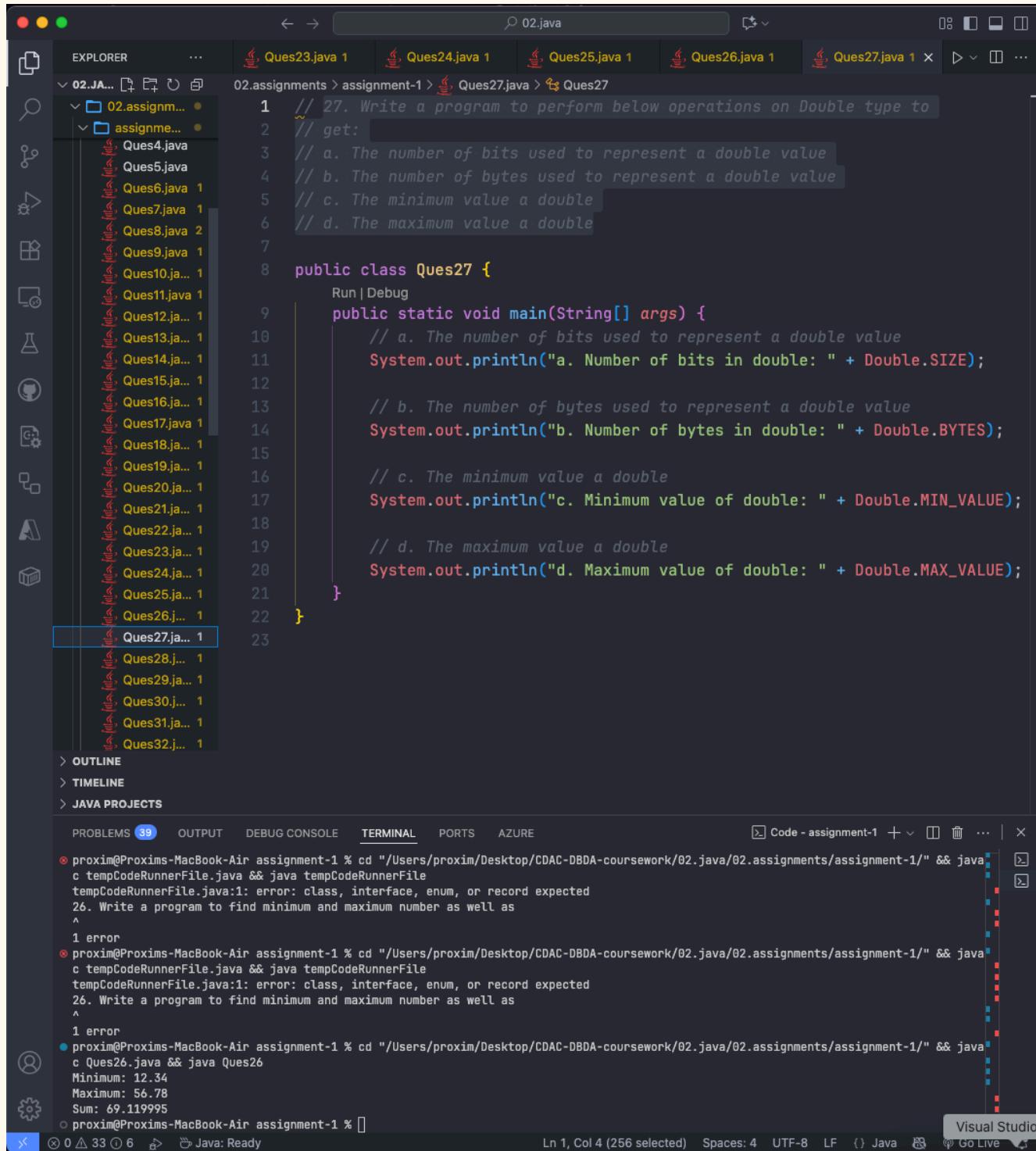
The code editor has a sidebar with a list of Java files, including Ques22.java, Ques23.java, Ques24.java, Ques25.java, and Ques26.java. The Ques26.java file is the active tab. The bottom of the screen shows the terminal output, which shows the program being run and the results: Minimum: 12.34, Maximum: 56.78, and Sum: 69.119995.

PROBLEMS 39 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
① proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
26. Write a program to find minimum and maximum number as well as
^
1 error
② proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c tempCodeRunnerFile.java && java tempCodeRunnerFile
tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
26. Write a program to find minimum and maximum number as well as
^
1 error
③ proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
c Ques26.java && java Ques26
Minimum: 12.34
Maximum: 56.78
Sum: 69.119995
④ proxim@Proxims-MacBook-Air assignment-1 %
```

Ln 6, Col 26 Spaces: 4 UTF-8 LF ( Java Go Live

## Ques 27.



Code - assignment-1

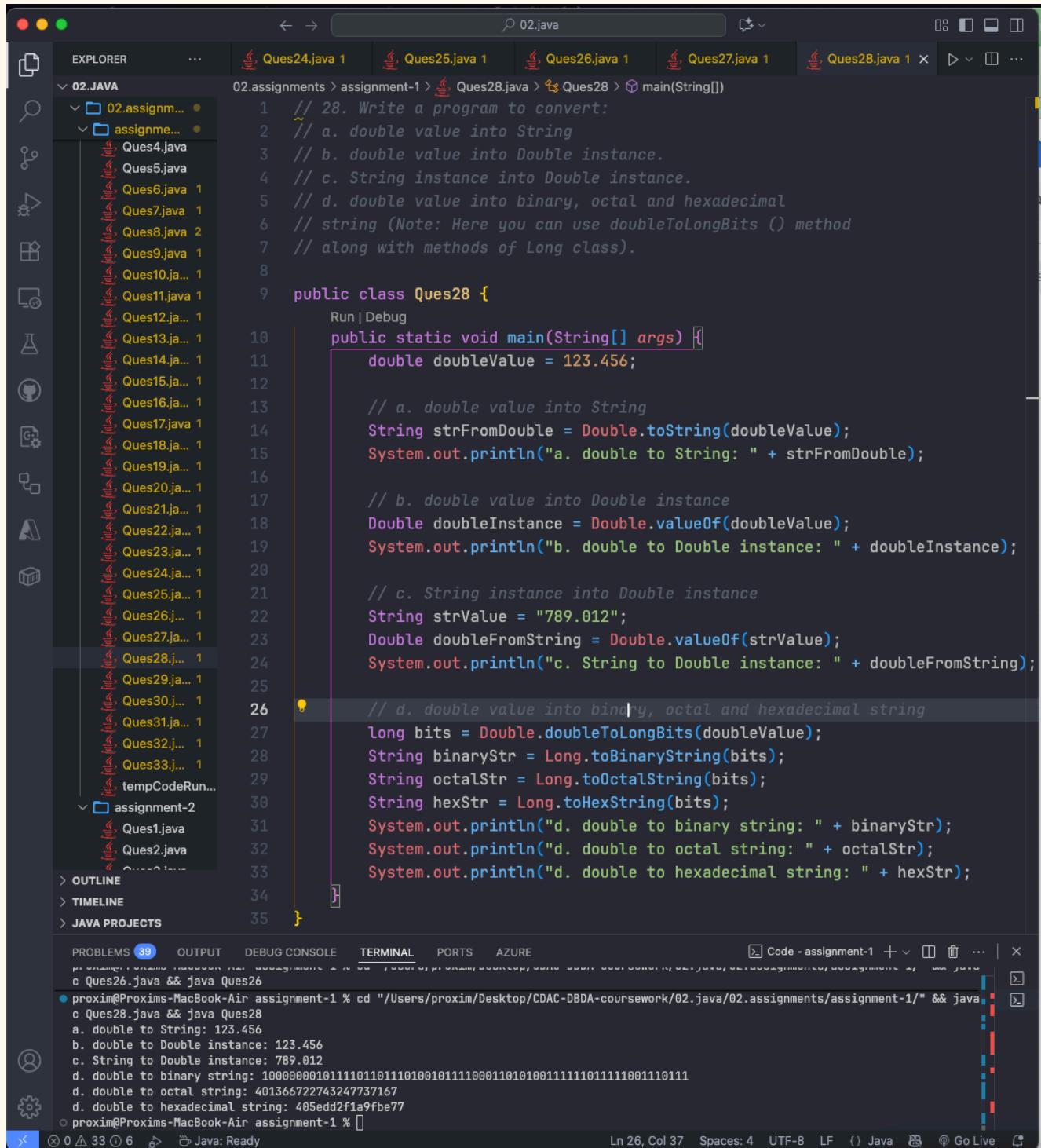
```
1 // 27. Write a program to perform below operations on Double type to
2 // get:
3 // a. The number of bits used to represent a double value
4 // b. The number of bytes used to represent a double value
5 // c. The minimum value a double
6 // d. The maximum value a double
7
8 public class Ques27 {
9     Run | Debug
10    public static void main(String[] args) {
11        // a. The number of bits used to represent a double value
12        System.out.println("a. Number of bits in double: " + Double.SIZE);
13
14        // b. The number of bytes used to represent a double value
15        System.out.println("b. Number of bytes in double: " + Double.BYTES);
16
17        // c. The minimum value a double
18        System.out.println("c. Minimum value of double: " + Double.MIN_VALUE);
19
20        // d. The maximum value a double
21        System.out.println("d. Maximum value of double: " + Double.MAX_VALUE);
22    }
23 }
```

PROBLEMS 39 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
✖ proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
  c tempCodeRunnerFile.java && java tempCodeRunnerFile
  tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
  26. Write a program to find minimum and maximum number as well as
  ^
  1 error
✖ proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
  c tempCodeRunnerFile.java && java tempCodeRunnerFile
  tempCodeRunnerFile.java:1: error: class, interface, enum, or record expected
  26. Write a program to find minimum and maximum number as well as
  ^
  1 error
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java
  c Ques26.java && java Ques26
  Minimum: 12.34
  Maximum: 56.78
  Sum: 69.119995
○ proxim@Proxims-MacBook-Air assignment-1 %
```

Ln 1, Col 4 (256 selected) Spaces: 4 UTF-8 LF () Java Visual Studio Go Live

## Ques 28.



02.assignments > assignment-1 > Ques28.java > Ques28 > main(String[])

```
1 // 28. Write a program to convert:
2 // a. double value into String
3 // b. double value into Double instance.
4 // c. String instance into Double instance.
5 // d. double value into binary, octal and hexadecimal
6 // string (Note: Here you can use doubleToLongBits () method
7 // along with methods of Long class).
8
9 public class Ques28 {
10     Run|Debug
11     public static void main(String[] args) {
12         double doubleValue = 123.456;
13
14         // a. double value into String
15         String strFromDouble = Double.toString(doubleValue);
16         System.out.println("a. double to String: " + strFromDouble);
17
18         // b. double value into Double instance
19         Double doubleInstance = Double.valueOf(doubleValue);
20         System.out.println("b. double to Double instance: " + doubleInstance);
21
22         // c. String instance into Double instance
23         String strValue = "789.012";
24         Double doubleFromString = Double.valueOf(strValue);
25         System.out.println("c. String to Double instance: " + doubleFromString);
26
27         // d. double value into binary, octal and hexadecimal string
28         long bits = Double.doubleToLongBits(doubleValue);
29         String binaryStr = Long.toBinaryString(bits);
30         String octalStr = Long.toOctalString(bits);
31         String hexStr = Long.toHexString(bits);
32
33         System.out.println("d. double to binary string: " + binaryStr);
34         System.out.println("d. double to octal string: " + octalStr);
35         System.out.println("d. double to hexadecimal string: " + hexStr);
}
```

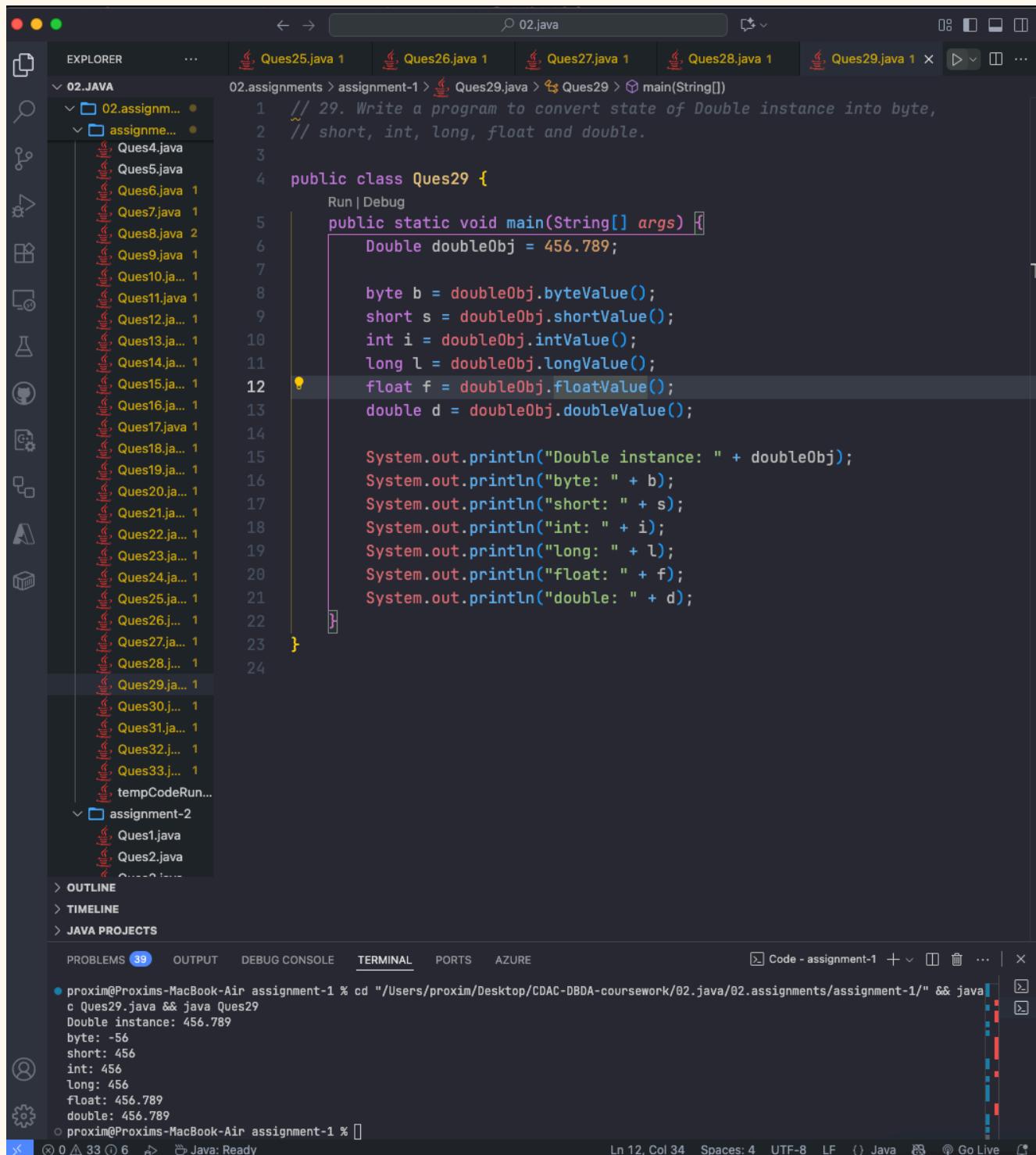
PROBLEMS 39 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Code - assignment-1

```
c Ques26.java && java Ques26
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques28
c Ques28.java && java Ques28
a. double to String: 123.456
b. double to Double instance: 123.456
c. String to Double instance: 789.012
d. double to binary string: 10000000101110101110100110101001111101111100111011
d. double to octal string: 401366722743247737167
d. double to hexadecimal string: 405eddd2f1a9fbe77
```

Ln 26, Col 37 Spaces: 4 UTF-8 LF () Java Go Live

## Ques 29.



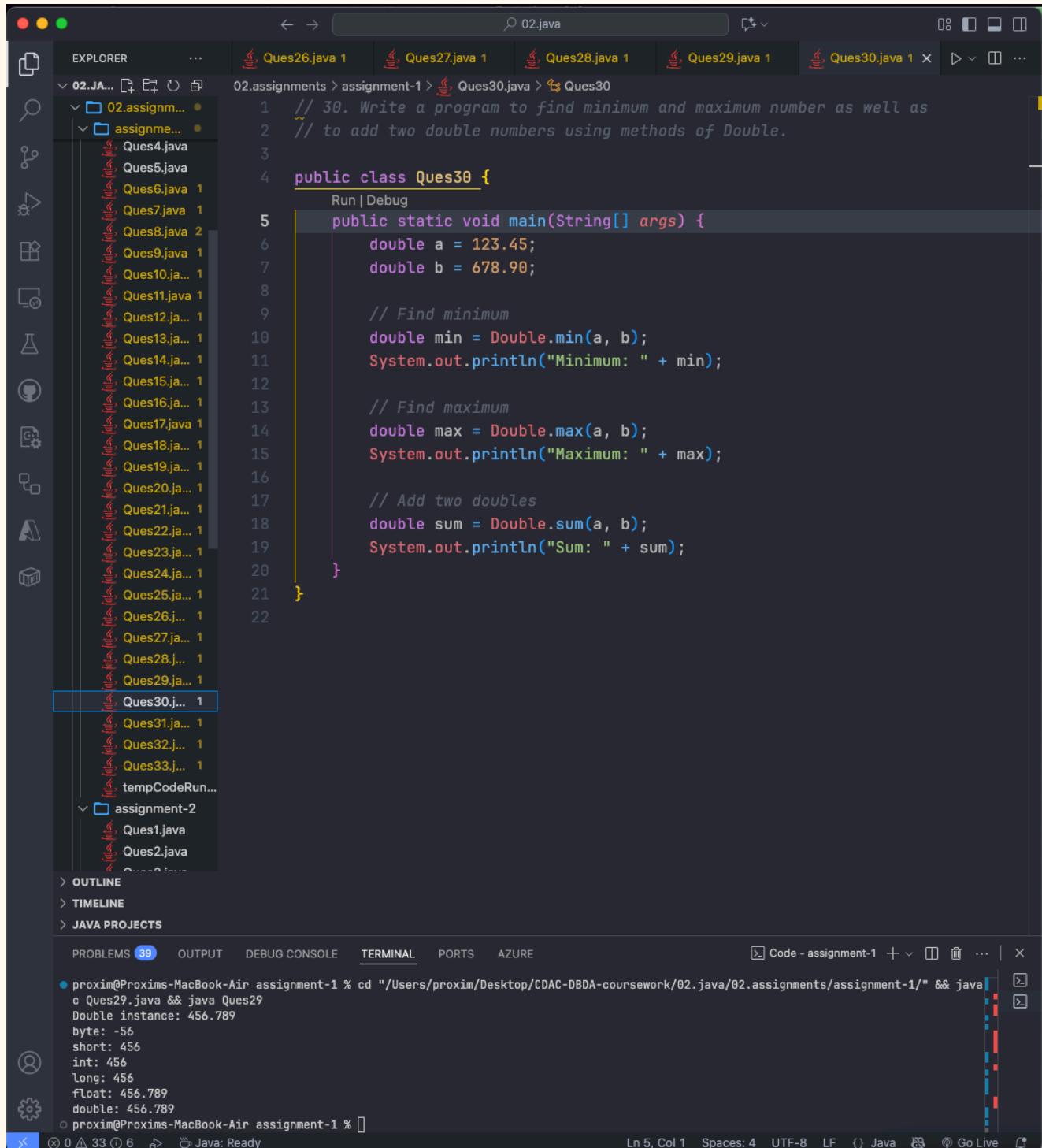
The screenshot shows a Java code editor with the following details:

- File:** 02.java
- Code Content:**

```
1 // 29. Write a program to convert state of Double instance into byte,
2 // short, int, long, float and double.
3
4 public class Ques29 {
5     Run | Debug
6     public static void main(String[] args) {
7         Double doubleObj = 456.789;
8
9         byte b = doubleObj.byteValue();
10        short s = doubleObj.shortValue();
11        int i = doubleObj.intValue();
12        long l = doubleObj.longValue();
13        float f = doubleObj.floatValue();
14        double d = doubleObj.doubleValue();
15
16        System.out.println("Double instance: " + doubleObj);
17        System.out.println("byte: " + b);
18        System.out.println("short: " + s);
19        System.out.println("int: " + i);
20        System.out.println("long: " + l);
21        System.out.println("float: " + f);
22        System.out.println("double: " + d);
23    }
24}
```
- Explorer:** Shows a list of Java files in the 02.JAVA folder, including Ques25.java, Ques26.java, Ques27.java, Ques28.java, and Ques29.java.
- Terminal:** Shows the command line output of running the program:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques29
Double instance: 456.789
byte: -56
short: 456
int: 456
long: 456
float: 456.789
double: 456.789
```
- Status Bar:** Shows the line and column number (Ln 12, Col 34), the number of spaces (Spaces: 4), the character encoding (UTF-8), the line separator (LF), the Java language version (Java), and the Go Live button.

## Ques 30.



The screenshot shows a Java code editor within an IDE. The top bar displays the file name "02.java". The left sidebar shows the "EXPLORER" view with a tree structure of Java files. The "assignment-1" folder contains "Ques30.java" and other files like "Ques26.java 1" through "Ques29.java 1". The "assignment-2" folder contains "Ques1.java" and "Ques2.java". The main editor area shows the code for "Ques30.java". The code is as follows:

```
// 30. Write a program to find minimum and maximum number as well as
// to add two double numbers using methods of Double.

public class Ques30 {
    public static void main(String[] args) {
        double a = 123.45;
        double b = 678.90;

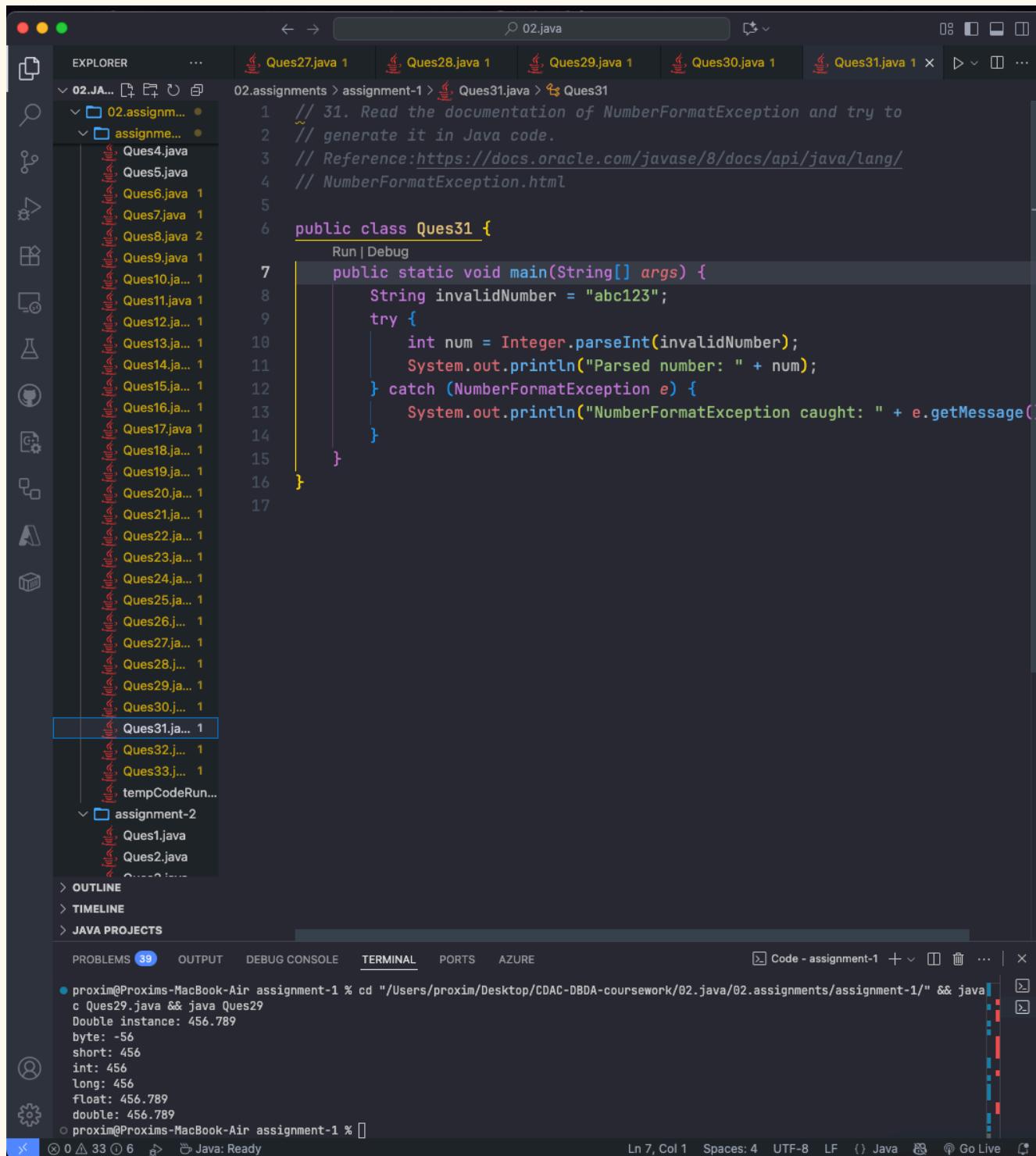
        // Find minimum
        double min = Double.min(a, b);
        System.out.println("Minimum: " + min);

        // Find maximum
        double max = Double.max(a, b);
        System.out.println("Maximum: " + max);

        // Add two doubles
        double sum = Double.sum(a, b);
        System.out.println("Sum: " + sum);
    }
}
```

The "PROBLEMS" tab in the bottom left shows a single error: "Ques29.java && java Ques29". The "TERMINAL" tab shows the command "cd /Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/ && java Ques29" and its output, which includes various primitive type definitions. The status bar at the bottom indicates "Java: Ready".

## Ques 31.



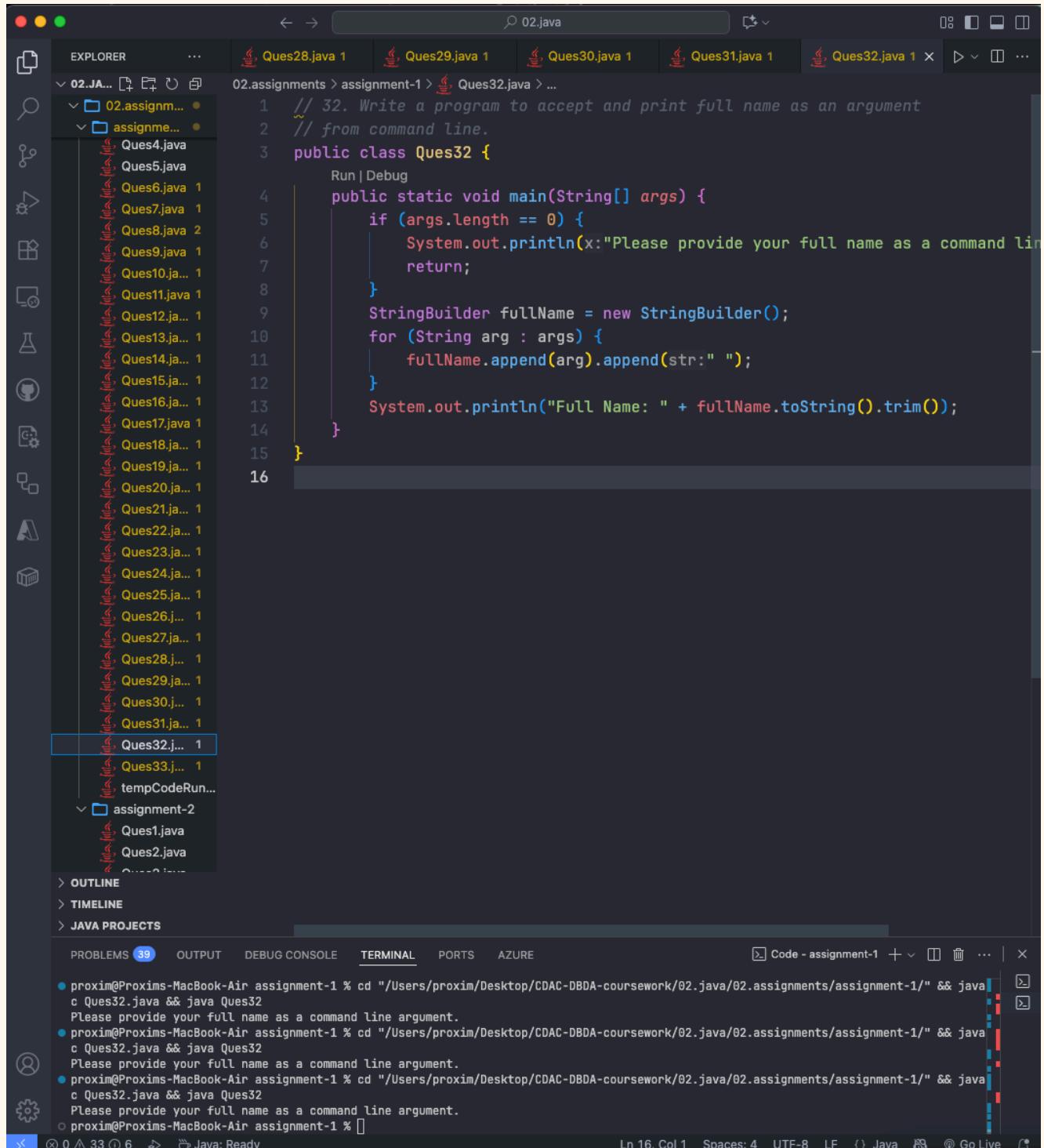
The screenshot shows a Java IDE interface with the following details:

- File Explorer:** Shows a tree structure of Java files. The current file is `Ques31.java` under the `02.assignments/assignment-1` folder.
- Code Editor:** Displays the code for `Ques31.java`. The code is a simple demonstration of handling a `NumberFormatException` by printing an invalid number to the console.
- Terminal:** Shows the command-line output of running the code. The output shows the conversion of the string "abc123" to an integer, resulting in a `NumberFormatException` with the message "java: invalid number".
- Status Bar:** Shows the current Java version is "Java: Ready".

```
1 // 31. Read the documentation of NumberFormatException and try to
2 // generate it in Java code.
3 // Reference:https://docs.oracle.com/javase/8/docs/api/java/lang/
4 // NumberFormatException.html
5
6 public class Ques31 {
7     public static void main(String[] args) {
8         String invalidNumber = "abc123";
9         try {
10             int num = Integer.parseInt(invalidNumber);
11             System.out.println("Parsed number: " + num);
12         } catch (NumberFormatException e) {
13             System.out.println("NumberFormatException caught: " + e.getMessage());
14         }
15     }
16 }
17
```

```
● proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques31
Double instance: 456.789
byte: -56
short: 456
int: 456
long: 456
float: 456.789
double: 456.789
● proxim@Proxims-MacBook-Air assignment-1 %
```

## Ques 32.



The screenshot shows a Java code editor with the following details:

- File Explorer:** Shows a file structure under "02.JA...". The "assignment-1" folder contains "Ques32.java" (selected), "Ques28.java", "Ques29.java", "Ques30.java", "Ques31.java", and "Ques32.java" (closed). The "assignment-2" folder contains "Ques1.java" and "Ques2.java".
- Code Editor:** The "Ques32.java" file contains the following code:

```
1 // 32. Write a program to accept and print full name as an argument
2 // from command line.
3 public class Ques32 {
4     Run | Debug
5     public static void main(String[] args) {
6         if (args.length == 0) {
7             System.out.println("Please provide your full name as a command line argument");
8             return;
9         }
10        StringBuilder fullName = new StringBuilder();
11        for (String arg : args) {
12            fullName.append(arg).append(" ");
13        }
14        System.out.println("Full Name: " + fullName.toString().trim());
15    }
16 }
```
- Terminal:** The terminal shows the output of running the program:

```
proxim@Proxims-MacBook-Air assignment-1 % cd "/Users/proxim/Desktop/CDAC-DBDA-coursework/02.java/02.assignments/assignment-1/" && java Ques32
Please provide your full name as a command line argument.
```
- Bottom Status:** Status bar shows "Ln 16, Col 1" and "Java: Ready".

Ques 33.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a file tree with a folder structure under "02.JA...". The "02.assignm..." folder is expanded, showing files like Ques4.java, Ques5.java, Ques6.java, Ques7.java, Ques8.java, Ques9.java, Ques10.java, Ques11.java, Ques12.java, Ques13.java, Ques14.java, Ques15.java, Ques16.java, Ques17.java, Ques18.java, Ques19.java, Ques20.java, Ques21.java, Ques22.java, Ques23.java, Ques24.java, Ques25.java, Ques26.java, Ques27.java, Ques28.java, Ques29.java, Ques30.java, Ques31.java, Ques32.java, and Ques33.java. The "Ques33.java" file is selected and highlighted with a blue border.
- Code Editor:** The main editor area displays the Java code for "Ques33.java". The code handles command-line arguments to perform arithmetic operations (addition, subtraction, multiplication, division) on an integer, a float, and a double.
- Terminal:** The bottom terminal window shows the execution of the code. It prompts for a command-line argument, then displays the usage information, and finally performs a division operation (3 / 2.0) resulting in 1.5.