1. What are the two types of errors?

Compilation errors and runtime errors (exceptions)

2. What are the three ways to deal with exceptions in Java?

1. Throw them to the previous method call

2. Surround them with a try-catch-statement

3. Write code to avoid the situation to begin with

3. What is a stack trace, and what does it tell you?

A stack trace is an error message that tells you the list of method calls that lead to the exception that was thrown, what the exception was, the message from the exception, and the line that caused it.

4. Write out the Java syntax for a try-catch-statement:

try {

// code that could throw an exception

} catch(<type\_of\_exception> <name\_of\_exception>) {

// code that runs after an exception is thrown

}

5. What is the most common exception you will get in Java, and what causes it?

NullPointerException, it is caused when you try to call a method or access a variable from a null value

6. Connect each exception to the code snippet that would cause it to be thrown:

NumberFormatException

InputMismatchException

NullPointerException

FileNotFoundException

IllegalArgumentException

ArrayIndexOutOfBoundsException

int age = input.nextInt();

setHeight(-5.0);

System.out.println(arr[-1]);

String s = null; s.substring(3, 5);

FileReader fr = new FileReader(“data.txt”);

Integer.parseInt(“13.5”);

7. Modify the following code snippets to avoid an exception:

public static boolean checkEqual(Object obj1, Object obj2){

// throws a NullPointerException if obj1 is null

if(obj1 == null)

return false;

return obj1.equals(obj2);

}

public static void setValue(int[] arr, int idx, int newValue){

// throws ArrayIndexOutOfBoundsException if idx < 0 or >= arr.length

if(idx < 0 || idx > arr.length)

return;

arr[idx] = newValue;

}

8. Surround the following code snippets with a try-catch-statement to avoid an exception:

public static void main(String[] args){

File file = new File(“data.txt”);

// could throw a FileNotFoundException

try{

FileReader fr = new FileReader(file);

} catch(FileNotFoundException e){

System.out.println(“File Not Found”);

}

}

public static int readNext(Scanner input){

// could throw an InputMismatchException

try{

return input.nextInt();

} catch(InputMismatchException e){

System.out.println(“You must enter an int”);

return -1;

}

}