1. (6 points) Suppose that statement2 causes an exception in the following try-catch block:

**public void m2() {**

m1();

}

**public void m1() {**

**try** {

statement1;

statement2;

statement3;

}

**catch** (Exception1 ex1) {

}

**catch** (Exception2 ex2) {

}

statement4;

}

Answer the following questions:

* Will statement3 be executed?
* If the exception is not caught, will statement4 be executed?
* If the exception is caught in the catch block, will statement4 be executed?

ANS:

Will statement3 be executed?

**Answer:** No.

If the exception is not caught, will statement4 be executed?

**Answer:** No.

If the exception is caught in the catch clause, will statement4 be executed?

**Answer:** Yes.

1. (4 points) Why does the following method have a compile error?

**public void** m(**int** value) {

**if** (value < 40)

**throw new** Exception("value is too small");

}

ANS:

The method throws a checked exception. You have to declare to throw the exception in the method header.

1. (4 points)Write Java statement(s) to open a file and to create and open an output stream to a file named autos.txt. Assume the file’s location is C:\autos.txt

ANS:

File file = new File(“C:\\autos.txt”);

PrintWriter outputStream = new PrintWriter(file);

1. (10 points) Create try and catch block that opens a file named statistics.txt for output. Writes the integers 24, 55, and 76 to the file, and then closes the file. Assume the file location is C:\statistics.txt

ANS:

PrintWriter outputFile = null;

try

{

outputFile = new PrintWriter("C:\\statistics.txt");

outputFile.println(24);

outputFile.println(55);

outputFile.println(76);

outputFile.close();

}

catch(FileNotFoundException e)

{

System.out.println("Error opening the file autos.txt");

System.exit(0);

}

1. (5 points)Write a complete Java program that prompts the user for two nonnegative integer numbers. Your program should handle bad input data without using try catch blocks. Hint: The exception is InputMismatchException.

ANS:

import java.util.\*;

public class GetNumbers2

{

public static void main(String args[]) throws InputMismatchException

{

Scanner stdin = new Scanner(System.in);

System.out.print("Enter the first non-negative number: ");

int firstNumber = stdin.nextInt();

System.out.print("Enter the second non-negative number: ");

int secondNumber = stdin.nextInt();

System.out.println("Your numbers are " + firstNumber + "and " +

secondNumber);

}

}

1. (5 points) Revise the program above to use a try/catch block to handle the exception.

ANS:

import java.util.\*;

public class GetNumbers2

{

public static void main(String args[])

{

Scanner stdin = new Scanner(System.in);

try

{

System.out.print("Enter the first non-negative number: ");

int firstNumber = stdin.nextInt();

System.out.print("Enter the second non-negative number: ");

int secondNumber = stdin.nextInt();

System.out.println("Your numbers are " + firstNumber + "and " +

secondNumber);

}

catch(InputMismatchException e)

{

System.out.println("Please enter a positive integer.");

}

}

}

1. (4 points) Write a Java statement that throws a new exception explicitly with the String: File Not Found.

ANS:

throw new Exception("File Not Found!");

1. (6 points) Use a catch block to display the exception thrown in problem 7 above.

ANS:

catch(Exception e)

{

System.out.println(e.getMessage());

System.exit(0);

}

1. (10 points)Define an exception class called FileNotFoundException. The class should have a constructor with no parameters. If an exception is thrown with this zero-argument constructor, getMessage should return “File Not Found!” The class should also have a constructor with a single parameter of type String. If an exception is thrown with this constructor, then getMessage returns the value that was used as an argument to the constructor.

ANS:

public class FileNotFoundException extends Exception

{

public FileNotFoundException()

{

super("File Not Found!");

}

public FileNotFoundException(String message)

{

super(message);

}

}

1. (4 points)If you were going to catch an exception of type Exception, where should this catch block be placed within your code?

ANS:

Catching an exception of type Exception is a very general exception, therefore it should be the last catch block in the series of catches, with the more specific exceptions being caught first.

1. (4 points) Should an application catch objects of type Error?

ANS:

Most programs should not attempt to catch errors, because the program usually cannot recover from such errors.

1. (True/False) (4 points)When an exception is thrown, the code in the surrounding try block continues executing and then the catch block begins execution.

ANS: False

1. (True/False) (4 points)The finally block contains code to be executed whether or not an exception is thrown in a try block.

ANS: True

1. (6 points) Consider the following code fragment.

int [] a = new int[50];

a[50] = 100;

Will this code fragment throw an exception? Answer Yes or No (2 points), then Explain(4 points).

Answer: This code will throw an ArrayIndexOutOfBoundsException since the indexes of a run from 0 to 49, and this code fragment tries to access index 50.

1. (6 points) Is the Exception class a subclass of the Error class? Answer Yes or No (2 points), then Explain(4 points).

Answer: No the Exception class is not a subclass of the Error class, because there is no *is-a* relationship between them. They are actually siblings in an inheritance hierarchy. They are both subclasses of the Throwable class.

1. (4 points) Which of the following exception types must always be caught unless they are contained in methods that throw them in the method header?

a) file stream

b) IO

c) checked

d) unchecked

e) none of the above

Answer: c

Explanation: Checked exceptions must always be caught or thrown, or else a compiler error will be generated.

1. (4 points) Which of the following file streams should be explicitly closed to ensure that written data is properly retained?

a) output

b) input

c) error

d) writable

e) readable

Answer: a

Explanation: Output file streams should be explicitly closed using the close method so that all data is properly retained.

1. (4 points) A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is used to specify how certain exceptions should be handled.

a) finally block

b) try block

c) catch block

d) error

e) none of the above

Answer: c

Explanation: A catch block specifies the actions to be performed when a particular exception is thrown in a try block.

1. (4 points) A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ can be used to find the exact line where an exception was thrown during program execution.

a) interface

b) call-stack trace

c) try block

d) catch block

e) none of the above

Answer: b

Explanation: A call-stack indicates the exact line where an exception is thrown.