

Similar (but not same) tasks will be given in the midterm exam. Each person gets three programming tasks and about five simple questions.

**Question:** Is it okay that I prepare template programs for the midterm exam.

**Answer:** Yes, and it is encouraged.

**Sample Programming Task 1 [25 points]** – The purpose is to use Exception Handling, read from a text file. Able to use BigInteger class.

The program allows the user to enter a URL address, read the content from this user specified URL (if it is an invalid URL, handle this exception **MalformedURLException** and output “invalid URL, please enter a valid one.”).

Template <https://github.com/YalingZheng/CIS368Spr20/blob/master/midterm/sampleReadFile.java>

The program then read the numbers (are integer numbers such as 100; if a non-number is met, handle the exception by skipping this number), and multiple them. The multiplication result could be a big number, so please use BigInteger class for the multiplication result.

The program finally output the result.

Please submit the java file, snapshot of the console output to the CSU Blackboard.

## Sample Programming Task 2 [25 points]

Write a digital time clock. It shows the digital clock (hour 0 to 12, minute 00 to 59, PM or AM, seconds 00 to 59) as below in the center of the window.

03: 00: 33 PM

The clock can be enlarged by pressing up key; The clock can be shrunk by pressing down key.

Please refer the code below to enlarge and shrink elements.

<https://github.com/YalingZheng/CIS368Spr20/blob/master/Chap15animations/animations/USMap.java>

This requires you to group elements together and enlarge them.

Please submit the java file, snapshot of the clock to the CSU Blackboard.

## Sample Programming Task 3 [25 points in total]

(Add and remove points). Write a program that let the user click on a pane to dynamically create and remove points. When the user left-click the mouse (primary button), a point is created and displayed at the mouse point. The user can remove a point by pointing to it and right-clicking the mouse (secondary button).

Sample answer <http://soulutionmanual.blogspot.com/2016/12/chapter-15-exercise-15-introduction-to.html>

Please submit the java file, snapshot of the console output and result text output file to the CSU Blackboard.

### Sample Programming Task 4 [25 points]

The program allows the user to enter a URL address, read the content from this user specified URL (if it is an invalid URL, handle this exception **MalformedURLException** and output "invalid URL, please enter a valid one.").

The program then read telephone numbers (non telephone format numbers are ignored), output valid phone numbers and save the valid phone numbers to a local text file.

Please submit the java file, snapshot of the console output and result text output file to the CSU Blackboard.

### Sample Questions 5 [25 points]

Please review the checkpoint questions in each chapter; the questions will be from the checkpoint questions. If you review, it will take you less time to answer questions; otherwise, you will have to check textbook or Internet to get answers, which takes more time.

5.1 In the observer design pattern, which two interfaces are defined? Which methods are defined in these two interfaces?

5.2 Which of the following statements will throw an exception?

```
System.out.println(1 / 0);  
System.out.println(1.0 / 0);
```

5.3 What is a checked exception, and what is an unchecked exception?

5.4 Which of the following classes defines a legal abstract class?

(a)

```
class A {  
    abstract void unfinished() {  
    }  
}
```

(b)

```
public class abstract A {  
    abstract void unfinished();  
}
```

(c)

```
public class abstract A {  
    abstract void unfinished();  
}
```

}

(d)

```
abstract class A {  
    protected void unfinished();  
}
```

(e)

```
abstract class A {  
    abstract void unfinished();  
}
```

(f)

```
abstract class A {  
    abstract int unfinished();  
}
```

5.5 Which of the following is a correct interface?

(a)

```
interface A {  
    void print() { }  
}
```

(b)

```
abstract interface A {  
    abstract void print() { }  
}
```

(c)

```
abstract interface A {  
    print();  
}
```

(d)

```
interface A {  
    void print();  
}
```

(e)

```
interface A {  
    default void print() {  
    }  
}
```

(f)

```
interface A {  
    static int get() {  
        return 0;  
    }  
}
```