**Signature** _____ **CSE 8B** **Name** _____

**Quiz 5**

**cs8b** ____ **Winter 2015** **Student ID** _____

This quiz is to be taken **by yourself** with closed books, closed notes, no calculators.

Use the **letters** provided to fill in the blanks to define a class to handle button press events in JavaFX.

```
class MyButtonPressEventHandler implements ____D____ < ____M____ > {

  @Override
  public void ____F____ ( ____M____ e ) {
    System.out.println( "Button Pressed" );
  }
}
```

| | | |
|---|---|---|
| A) ButtonHandler<br>B) ButtonPressHandler<br>C) PressHandler<br>D) EventHandler<br>E) ActionHandler | F) handle<br>G) handleButtonPress<br>H) handleButtonEvent<br>I) handlePress<br>J) handleEvent | K) ButtonEvent<br>L) ButtonPressEvent<br>M) ActionEvent<br>N) ActionPress<br>O) ButtonAction |

How do you register the above event handler with a Button?

```
Button button = new Button( "Press Me" );
MyButtonPressEventHandler handler1 = new MyButtonPressEventHandler();

button.____D____( handler1 );
```

A) setOnPress
B) setOnEvent
C) setOnHandler
D) setOnAction
E) setOnButtonPress

What is true with having an event handler defined as a member inner class of the GUI class that is defining the GUI components this event handler is registered to handle events on? _____

A) Event handler class can be defined private
B) Event handler code can directly access any of the private members of the outer GUI class
C) Event handler class cannot be defined private
D) Event handler code cannot directly access any of the private members of the outer GUI class
E) Both A and B
F) Both C and D

In the Java graphics coordinate system, where is (0,0)? ____C____

A) Center of the pane
B) Upper right corner
C) Upper left corner
D) Lower right corner
E) Lower left corner

Using the code segment below, answer the questions in the question boxes to the right. **Write the letter** of the correct answer in the space provided for each question.

```
try {
  // Block A
}
catch ( ExceptionType1 e1 ) {
  // Block B
}
catch ( ExceptionType2 e2 ) {
  // Block C
}
finally {
  // Block D
}
// Block E
```

Which code block will always be executed whether or not an exception occurs or is caught? D

B) Block B  D) Block D
C) Block C  E) Block E

Which code block will execute the normal code which may cause an exception we want to catch? A

A) Block A  D) Block D
B) Block B  E) Block E
C) Block C

Which code block should have code to handle the most general exception type (higher in the hierarchy)? C

A) Block A  D) Block D
B) Block B  E) Block E
C) Block C

Under what circumstance(s) will code block E be executed? _____

A) No exception occurs
B) Exception occurs and is not caught
C) Exception occurs and is caught with no return statement in catch block
D) A&B  F) B&C
E) A&C  G) A&B&C

Fill in the missing parts in the class designed to be used with threads below

```
public class Computation implements Runnable {
  private final int startRange;
  private final int endRange;
  private final byte[] array;
  private long mySumSquared;

  public Computation( int start, int end, byte[] arr ) {
    this.startRange = start;  this.endRange = end;
    this.array = arr;  this.mySumSquared = 0;
  }

  @Override
  public void run() {
    for ( int i = startRange; i < endRange && i < array.length; i++ ) {
      mySumSquared += array[i]*array[i];
    }
    System.out.println( "Sum Squared = " + mySumSquared );
  }
}
```

Write the code to create a **single** thread which executes the Computation object above and to get that thread running.

```
byte[] arr = // code to create array here.
int start_val = 0;
int end_val = arr.length;

Computation c = new Computation(start_val, end_val, arr) ;

Thread t = new Thread(c) ;

t.start() ;
```