

Student Name: _____ Student ID # _____

UOSA Statement of Academic Integrity

On my honor I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.

Signature: _____ Date: _____

Notes Regarding this Examination

Open Book(s) You may consult any printed textbooks in your immediate possession during the course of this examination.

Open Notes You may consult any printed notes in your immediate possession during the course of this examination.

No Electronic Devices Permitted You may not use any electronic devices during the course of this examination, including but not limited to calculators, computers, and cellular phones. All electronic devices in the student's possession must be turned off and placed out of sight (for example, in the student's own pocket or backpack) for the duration of the examination.

Violations Copying another's work, or possession of electronic computing or communication devices in the testing area, is cheating and grounds for penalties in accordance with school policies.

Question 1: Graphics (10 points)

Explain one reason that your code might call `paintComponent ()` directly.

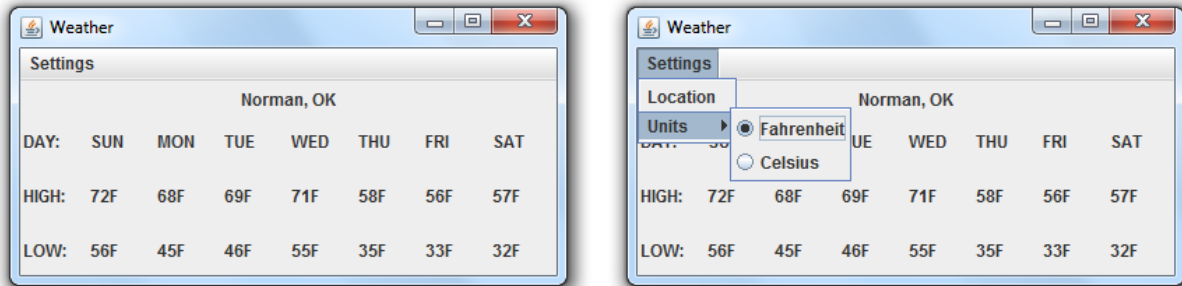
Explain one reason that your code might call `paintComponent ()` indirectly.

Question 2: Event Driven Programming (10 points)

Explain why `AbstractButton` (the parent class of both `JButton` and `JMenuItem`) contains a call to the constructor for `ActionEvent` while `JLabel` does not.

Question 3: Event Driven Programming Redux (10 points)

Explain why the various constructors for `ActionEvent` all take a parameter of type `String`.

Question 4: Graphical User Interfaces (20 points)

Explain how the GUI shown above could be constructed using common Java Swing components and containers and common Java AWT layout managers we have covered in class. Note that, as the overall window is enlarged or reduced, the rows and columns showing the days of the week and the high and low temperatures (including their labels in the left-hand column) grow or shrink but the other viewable items do not.

Question 5: MVC (30 points)

Nathalie wants to create a store inventory management system in Java following the MVC paradigm as we have discussed it in class and using data encapsulation best practices. The GUI should allow the user to enter data on current inventories, expected deliveries, current sales, and historic sales quantities. The program should use this data to create a list of orders to place to try to keep inventories at desired levels and display this list to the user and allow the user to specify a file to which this list is written. *Explain* what needs to go into each part of the code in order to accomplish all this.

A. Driver

B. Model

C. View

D. Controller

Question 6: MVC Revisited (20 points)

After Nathalie's program from Question 5 is completed and working according to specifications, Mohak wants to add functionality to it so that the user can edit the list of orders using the GUI before writing it to the file. To accomplish this, how will each part of the original program need to be modified? *Explain* each answer.

A. Driver

B. Model

C. View

D. Controller