Question 1. [3 MARKS]

Short Answer

Part (a) [1 MARK]
When writing a class, if you want it to contain a variable that keeps track of the number of instances of that class, would you make that variable static or non-static?

Answer: ____________________________________________

Part (b) [1 MARK]
Given a variable \( v \) that points to an instance of a class \( C \), is it possible to access a public static variable of \( C \) using an expression that starts with \( v \)?

Answer (circle one): YES NO

Part (c) [1 MARK]
When writing a subclass \( \text{Sub} \) of a class \( \text{Sup} \), is it possible to customize (override) the behaviour of a static method of \( \text{Sup} \)?

Answer (circle one): YES NO
Question 2. [6 marks]

Suppose we are executing a Java program, and at this moment the memory model looks as shown below. (The contents of the static space is not included because it is not relevant.)

Circle the best answer to each of the following multiple-choice questions.

Part (a) [1 mark] To access P’s y using a2, I need to do
(1) a2.y  (2) a2.(P)y  (3) ((P) a2).y  (4) a2.super.y
(5) None of the above work, but it can be done.
(6) It cannot be done.

Part (b) [1 mark] To access C’s y using a3, I need to do
(1) a3.y  (2) a3.this.y  (3) a3.(C)y  (4) ((C) a3).y
(5) None of the above work, but it can be done.
(6) It cannot be done.

Part (c) [1 mark] To call P’s method r using a2, I need to do
(1) a2.r()  (2) a2.(P).r()  (3) ((P) a2).r()  (4) a2.super.r()
(5) None of the above work, but it can be done.
(6) It cannot be done.

Part (d) [1 mark] To access C’s method r using a3, I need to do
(1) a3.r()  (2) a3.this.r()  (3) a3.(C)r()  (4) ((C) a3).r()
(5) None of the above work, but it can be done.
(6) It cannot be done.

Part (e) [2 marks]

In the picture, draw the results of executing the following two statements. If values change, cross them out and write in the new values.

```java
a3 = a1;
a3.y = ((P) a2).y
```
Question 3.  [5 marks]

Listeners
In lecture, we discussed the following class.

```java
public class MooingWindow extends JFrame implements ActionListener {
    MooingWindow(String title) {
        super(title);
        JButton myButton = new JButton("Click to moo");
        myButton.addActionListener(this);
        JPanel contents = new JPanel();
        contents.add(myButton);
        this.setContentPane(contents);
        this.pack();
    }

    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(this, "Moo");
    }
}

public class Main3 {
    public static void main(String[] args) {
        MooingWindow mw1 = new MooingWindow("Moo Window One");
        mw1.setVisible(true);
    }
}
```

Part (a)  [1 MARK] What does the super call in the constructor do?

Part (b)  [4 MARKS]
This window is a listener for the button. If we were instead to write a separate class BListener to be the listener, what would need to change? You are encouraged to answer in point form.
This page is for rough work and for answers that didn’t fit in the space provided.