

UBIT: _____@buffalo.edu

Name: _____

CSE116 - Fall 2019

Level 2 Quiz

Q1. Inheritance and Polymorphism [15 points]

```
abstract class Crystal(var size: Double) {
  def value(): Double
  def grow(): Unit = {
    this.size += 1.0
  }
}

class Quartz(quartzSize: Double) extends Crystal(quartzSize) {
  override def value(): Double = {
    this.size * 10.0
  }
}

class Salt(saltSize: Double) extends Crystal(saltSize) {
  override def value(): Double = {
    this.size * 2.0
  }
  override def grow(): Unit = {
    this.size += 4.0
  }
}

object Q1 {
  def totalValue(crystals: List[Crystal]): Double = {
    var value = 0.0
    for(crystal <- crystals){
      value += crystal.value()
    }
    value
  }

  def growAll(crystals: List[Crystal]): Unit = {
    for(crystal <- crystals){
      crystal.grow()
    }
  }

  def main(args: Array[String]): Unit = {
    val quartz: Quartz = new Quartz(8.0)
    val salt: Salt = new Salt(4.0)
    val crystals: List[Crystal] = List(quartz, salt, new Quartz(1.0))
    totalValue(crystals) // part a
    growAll(crystals)
    totalValue(crystals) // part b
  }
}
```

What is the return value of `totalValue(crystals)` marked as part a? [5 points]

What is the return value of `totalValue(crystals)` marked as part b? [10 points]

Continue on back

Q2. State Pattern [20 points]

```
class VendingMachine {
  var state: State = new NoCoins(this)
  def insertCoins(): Unit = {
    this.state.insertCoins()
  }
  def pressCoinReturn(): Unit = {
    this.state.pressCoinReturn()
  }
  def pressBeverageButton(): Unit = {
    this.state.pressBeverageButton()
  }
  def destroy(): Unit = {
    this.state.destroy()
  }
}

abstract class State {
  def insertCoins(): Unit = {}
  def pressCoinReturn(): Unit = {}
  def pressBeverageButton(): Unit = {}
  def destroy(): Unit = {}
}

class PaidInFull(machine: VendingMachine) extends State {
  override def pressCoinReturn(): Unit = {
    machine.state = new NoCoins(machine)
  }
  override def pressBeverageButton(): Unit = {
    println("beverage dispensed")
    machine.state = new NoCoins(machine)
  }
  override def destroy(): Unit = {
    machine.state = new Destroyed(machine)
  }
}

class NoCoins(machine: VendingMachine) extends State {
  override def insertCoins(): Unit = {
    machine.state = new PaidInFull(machine)
  }
  override def pressBeverageButton(): Unit = {
    println("please insert coins")
  }
  override def destroy(): Unit = {
    machine.state = new Destroyed(machine)
  }
}

class Destroyed(machine: VendingMachine) extends State {
  override def destroy(): Unit = {
    println("what have I done to you?")
  }
}

object Q2 {
  def main(args: Array[String]): Unit = {
    val vendingMachine: VendingMachine = new VendingMachine()
    vendingMachine.pressBeverageButton()
    vendingMachine.insertCoins()
    vendingMachine.pressBeverageButton()
    vendingMachine.destroy()
    vendingMachine.insertCoins()
    vendingMachine.pressBeverageButton()
  }
}
```

What is printed to screen when this program (object Q2) executes? [5 points]

Draw the state diagram for the VendingMachine class. Include all states and state transitions along with the function called for each transition. [15 points]

Q3. GUI and MVC [15 points]

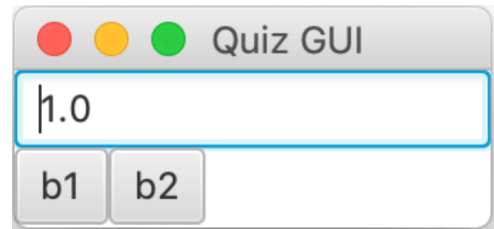
```
class Model {
  var number: Int = 1
  def displayNumber(): Double = {
    this.number
  }
  def firstApiMethod(magnitude: Int): Unit = {
    this.number += magnitude
  }
  def secondApiMethod(): Unit = {
    this.number *= 2
  }
}

class KeyEventHandler(model: Model) extends EventHandler[KeyEvent] {
  override def handle(event: KeyEvent): Unit = {
    event.getCode.getName match {
      case "A" => model.firstApiMethod(2)
      case "B" => model.secondApiMethod()
    }
  }
}

class OneButtonListener(model: Model) extends EventHandler[ActionEvent] {
  override def handle(event: ActionEvent): Unit = {
    model.firstApiMethod(10)
  }
}

class TwoButtonListener(model: Model) extends EventHandler[ActionEvent] {
  override def handle(event: ActionEvent): Unit = {
    model.secondApiMethod()
  }
}

object View extends JFXApp {
  val model: Model = new Model()
  val textField: TextField = new TextField
  val button1: Button = new Button {
    text = "b1"
    onAction = new OneButtonListener(model)
  }
  val button2: Button = new Button {
    text = "b2"
    onAction = new TwoButtonListener(model)
  }
  stage = new PrimaryStage {
    title = "Quiz GUI"
    scene = new Scene() {
      content = List(new GridPane {
        add(textField, 0, 0, 2, 1)
        add(button1, 0, 1)
        add(button2, 1, 1)
      })
    }
  }
  addEventFilter(KeyEvent.KEY_PRESSED, new KeyEventHandler(model))
  addEventFilter(Event.ANY, (_: Event) => textField.text.value = model.displayNumber().toString)
}
```



Assume the program is restarted before each of the 3 questions.

What is the value of `model.displayNumber()` after the user presses the buttons b1, b1, then b2?

What is the value of `model.displayNumber()` after the user presses the keys B, A, then B?

What is the value of `model.displayNumber()` after the user presses b2, B, A, then b1?