

Lab Quiz 2 (Section B10)

Full Name: _____

UBIT: _____

Student Instructions

- This is an exam environment
 - No talking
 - No material allowed except blank paper and writing utensils
 - Only ask TA's questions clarifying what a question is asking

Q1) Read and understand the code below

```
abstract class Employee(var salary: Double) {

    def pay(): Double

    def jobReview(): Unit = {
        this.salary += 5000.0
    }
}

class SoftwareEngineer(sal: Double) extends Employee(sal) {

    override def pay(): Double = {
        this.salary / 26.0
    }
}

class Janitor(sal: Double) extends Employee(sal) {

    override def pay(): Double = {
        this.salary / 26.0
    }

    override def jobReview(): Unit = {
        this.salary += 1000.0
    }
}

def accumulate(Employees: List[Employee]): Double = {
    var total = 0.0
    for(employee <- Employees){
        total += employee.pay()
    }
    total
}

def performJobReviews(Employees: List[Employee]): Unit = {
    for(employee <- Employees){
        employee.jobReview()
    }
}
```

Part 1) Given the code below, write what is printed to the screen.

```
def main(args: Array[String]): Unit = {  
    val softwareEngineer: SoftwareEngineer = new SoftwareEngineer(104000)  
    val janitor: Janitor = new Janitor(39000)  
  
    println(softwareEngineer.pay())  
    println(janitor.pay())  
    softwareEngineer.jobReview()  
    println(softwareEngineer.salary)  
    janitor.jobReview()  
    println(janitor.salary)  
    println(softwareEngineer.pay())  
}
```

Part 2) Once again, write what is printed to the screen.

```
def main(args: Array[String]): Unit = {  
    val employeeList: List[Employee] = List(new Janitor(39000), new  
    SoftwareEngineer(104000), new Janitor(52000), new SoftwareEngineer(130000))  
  
    println(accumulate(employeeList))    //What is printed here? 10 marks  
    performJobReviews(employeeList)  
  
    for(employee <- employeeList){  
        println(employee.salary)    // What is printed here? 15 marks  
    }  
}
```

Q2(50 points): Use the following MVC structured program to answer the questions below

ASSUME ALL NECESSARY IMPORTS HAVE BEEN MADE

```
class Model {  
    var number: Double = 0.0  
    def showNumber(): Double = {  
        this.number  
    }  
    def mapInput(magnitude: Double): Unit = {  
        this.number += Math.pow(Math.E, magnitude)  
    }  
    def clean(): Unit = {
```

```

        this.number = 0
    }
}
class Controller(model: Model) {
    def button1Pressed(event: ActionEvent): Unit = model.mapInput(1)
    def button2Pressed(event: ActionEvent): Unit = model.clean()
    def otherUserActions(event: KeyEvent): Unit = {
        event.getCode.getName match {
            case "0" => model.mapInput(0)
            case "1" => model.mapInput(1)
            case "2" => model.mapInput(2)
            case "3" => model.mapInput(3)
            case "4" => model.mapInput(4)
            case "5" => model.mapInput(5)
            case "6" => model.mapInput(6)
            case "7" => model.mapInput(7)
            case "8" => model.mapInput(8)
            case "9" => model.mapInput(9)
            case "`" => model.clean()
            case _ => model.mapInput(-2)
        }
    }
}
class QuizButton(display: String, action: EventHandler[ActionEvent])
    extends Button {

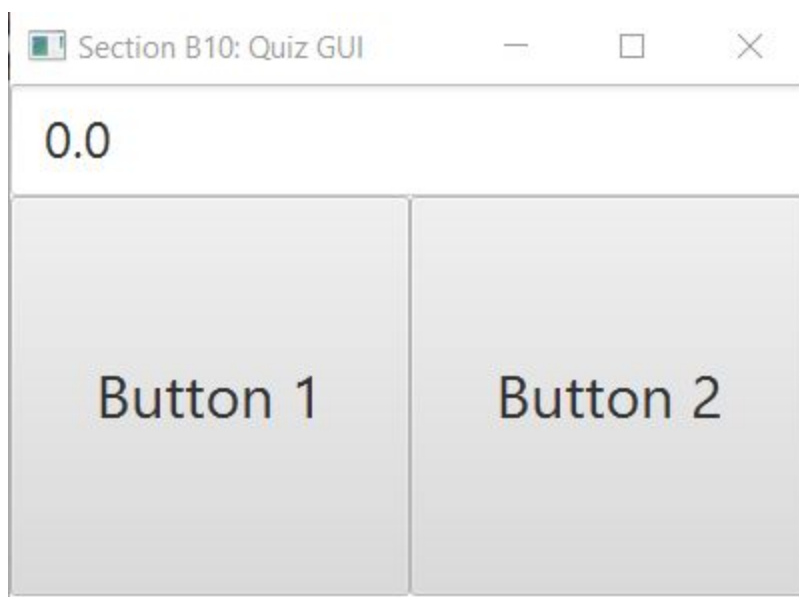
    val size = 200
    minWidth = size
    minHeight = size
    onAction = action
    text = display
    style = "-fx-font: 30 ariel;"
}
object View extends JFXApp {
    val model: Model = new Model()
    val controller: Controller = new Controller(model)
    var textField: TextField = new TextField {
        editable = false
        style = "-fx-font: 26 ariel;"
    }
}

```

```

        text.value = model.showNumber().toString
    }
    stage = new PrimaryStage {
        title = "Section B10: Quiz GUI"
        scene = new Scene() {
            content = List(
                new GridPane {
                    add(textField, 0, 0, 2, 1)
                    add(new QuizButton("Button 1", controller.button1Pressed),
0, 1)
                    add(new QuizButton("Button 2", controller.button2Pressed),
1, 1)
                }
            )
        }
        addEventFilter(KeyEvent.KEY_PRESSED,
controller.otherUserActions)
        // update the display after every event
        addEventFilter(Event.ANY, (event: Event) => textField.text.value
= model.showNumber().toString)
    }
}

```



What is displayed in the text field after each of the following sequences of user inputs?

Assume the program has been started just before the inputs for each of the 5 sequences (10 points each)

1. The user presses the button labelled b1
2. The user presses the buttons: b1, b1, b2, then b1
3. The user types "0000" on their keyboard
4. The user types "ABXRTDW" on their keyboard
5. The user presses the buttons: b1, b2, b1, b1 then types "000" on their keyboard