Lesson 4

Conditional Code: Checking for Switches

```
moveForward()
moveForward()

if isOnClosedSwitch {
    toggleSwitch()
}
moveForward()
if isOnClosedSwitch {
    toggleSwitch()
}
moveForward()
if isOnClosedSwitch {
    toggleSwitch()
}
```

Conditional Code: Using else if

```
moveForward()

if isOnClosedSwitch {
    toggleSwitch()
} else if isOnGem {
    collectGem()
}

moveForward()
if isOnClosedSwitch {
    toggleSwitch()
} else if isOnGem {
    collectGem()
}
```

Lesson 4

Conditional Code: Looping Conditional Code

```
for i in 1...13 {
moveForward()
    if isOnClosedSwitch {
        toggleSwitch()
    } else if isOnGem {
        collectGem()
    }
```

Conditional Code: Conditional Climb

```
for i in 1...13 {
    if isOnGem {
        collectGem()
        turnLeft()
        moveForward()
    } else {
        moveForward()
    }
}
```

Lesson 4

Conditional Code: Looping Conditional Code

```
for i in 1...13 {
moveForward()
    if isOnClosedSwitch {
        toggleSwitch()
    } else if isOnGem {
        collectGem()
    }
```

Conditional Code: Conditional Climb

```
for i in 1...13 {
    if isOnGem {
        collectGem()
        turnLeft()
        moveForward()
    } else {
        moveForward()
    }
}
```

Lesson 4

Conditional Code: Defining Smarter Functions

```
func collectOrToggle() {
    moveForward()
    moveForward()
    if isOnGem {
        collectGem()
    } else if isOnClosedSwitch {
        toggleSwitch()
    }
}
collectOrToggle()
collectOrToggle()
turnLeft()
moveForward()
moveForward()
turnLeft()
collectOrToggle()
collectOrToggle()
turnRight()
moveForward()
turnRight()
collectOrToggle()
collectOrToggle()
```

Lesson 4

Conditional Code: Defining Smarter Functions

```
func collectOrToggle() {
    moveForward()
    moveForward()
    if isOnGem {
        collectGem()
    } else if isOnClosedSwitch {
        toggleSwitch()
    }
}
collectOrToggle()
collectOrToggle()
turnLeft()
moveForward()
moveForward()
turnLeft()
collectOrToggle()
collectOrToggle()
turnRight()
moveForward()
turnRight()
collectOrToggle()
collectOrToggle()
```

Lesson 4

Conditional Code: Boxed In

```
func checkSquare() {
    if isOnGem {
        collectGem()
    } else if isOnClosedSwitch {
        toggleSwitch()
    }
}

func completeCorner() {
    checkSquare()
    moveForward()
    checkSquare()
    turnRight()
    moveForward()
}
```

```
moveForward()
turnRight()
for i in 1...4 {
    completeCorner()
}
```

Lesson 4

Conditional Code: Boxed In

```
func checkSquare() {
    if isOnGem {
        collectGem()
    } else if isOnClosedSwitch {
        toggleSwitch()
    }
}

func completeCorner() {
    checkSquare()
    moveForward()
    checkSquare()
    turnRight()
    moveForward()
}
```

```
moveForward()
turnRight()
for i in 1...4 {
    completeCorner()
}
```

Lesson 4

Conditional Code: Decision Tree

```
func solveRightSide() {
    turnRight()
    moveForward()
    moveForward()
    moveForward()
    turnLeft()
    moveForward()
    collectGem()
    turnLeft()
    moveForward()
    turnRight()
    moveForward()
    moveForward()
    moveForward()
    turnRight()
}
```

```
for i in 1…5 {
    moveForward()
    if isOnGem {
        solveRightSide()
    } else if isOnClosedSwitch {
        toggleSwitch()
        turnLeft()
        moveForward()
        collectGem()
        turnLeft()
        moveForward()
        turnLeft()
    }
```

Lesson 4

Conditional Code: Decision Tree

```
func solveRightSide() {
    turnRight()
    moveForward()
    moveForward()
    moveForward()
    turnLeft()
    moveForward()
    collectGem()
    turnLeft()
    moveForward()
    turnRight()
    moveForward()
    moveForward()
    moveForward()
    turnRight()
}
```

```
for i in 1…5 {
    moveForward()
    if isOnGem {
        solveRightSide()
    } else if isOnClosedSwitch {
        toggleSwitch()
        turnLeft()
        moveForward()
        collectGem()
        turnLeft()
        moveForward()
        turnLeft()
    }
```

Lesson 4

Logical Operators: Using the NOT Operator

```
for i in 1…4 {
    moveForward()
    if !isOnGem {
        turnLeft()
        moveForward()
        moveForward()
        collectGem()
        turnLeft()
        turnLeft()
        moveForward()
        moveForward()
        turnLeft()
    } else {
        collectGem()
    }
```

Logical Operators: Spiral of NOT

```
for i in 1...16 {
    if !isBlocked {
        moveForward()
    } else {
        turnLeft()
    }
}

toggleSwitch()
```

Lesson 4

Logical Operators: Using the NOT Operator

```
for i in 1…4 {
    moveForward()
    if !isOnGem {
        turnLeft()
        moveForward()
        moveForward()
        collectGem()
        turnLeft()
        turnLeft()
        moveForward()
        moveForward()
        turnLeft()
    } else {
        collectGem()
    }
```

Logical Operators: Spiral of NOT

```
for i in 1...16 {
    if !isBlocked {
        moveForward()
    } else {
        turnLeft()
    }
}

toggleSwitch()
```

Lesson 4

Logical Operators: Checking This AND That

```
for i in 1...7 {
    moveForward()
    if isOnGem && isBlockedLeft {
        turnRight
        moveForward()
        moveForward()
        collectGem()
        turnLeft()
        turnLeft()
        moveForward()
        moveForward()
        turnRight()
    } else if isOnGem {
        collectGem()
        moveForward()
    }
```

Logical Operators: Checking This OR That

```
for i in 1...12 {
    if isBlocked || isBlockedLeft {
        turnRight()
        moveForward()
    } else {
        moveForward()
    }
```

Lesson 4

Logical Operators: Checking This AND That

```
for i in 1...7 {
    moveForward()
    if isOnGem && isBlockedLeft {
        turnRight
        moveForward()
        moveForward()
        collectGem()
        turnLeft()
        turnLeft()
        moveForward()
        moveForward()
        turnRight()
    } else if isOnGem {
        collectGem()
        moveForward()
    }
```

Logical Operators: Checking This OR That

```
for i in 1...12 {
    if isBlocked || isBlockedLeft {
        turnRight()
        moveForward()
    } else {
        moveForward()
    }
```

Lesson 4

Logical Labryinth

```
for i in 1…6 {
    moveForward()
    if isOnClosedSwitch && isBlocked {
        toggleSwitch
        turnLeft()
        moveForward()
    } else if isOnClosedSwitch {
        toggleSwitch
        turnRight()
        moveForward()
        moveForward()
        collectGem()
        turnRight()
        turnRight()
        moveForward()
        moveForward()
        turnRight()
    }
```

***Some puzzles may have multiple solutions**

Lesson 4

Logical Labryinth

```
for i in 1…6 {
    moveForward()
    if isOnClosedSwitch && isBlocked {
        toggleSwitch
        turnLeft()
        moveForward()
    } else if isOnClosedSwitch {
        toggleSwitch
        turnRight()
        moveForward()
        moveForward()
        collectGem()
        turnRight()
        turnRight()
        moveForward()
        moveForward()
        turnRight()
    }
```

***Some puzzles may have multiple solutions**