

### A. Class Files (KOTLIN)

MainActivity:

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        lifecycle.addObserver(MyObserver())

        val viewModel by viewModels<MainViewModel>()

        viewModel.Qty.observe(this) {
            binding.txtQty.text = "QTY: $it"
        }

        viewModel.description.observe(this) {
            binding.txtDesc.text = "$it"
        }

        viewModel.message.observe(this) {
            binding.txtfee.text = "$it"
        }

        binding.bbtnpos.setOnClickListener {
            viewModel.apples(1)
        }
        binding.bbtnneg.setOnClickListener {
            viewModel.apples(2)
        }

        binding.bbtncheck.setOnClickListener() {
            viewModel.totalAmount()
            viewModel.total.observe(this) {
                Snackbar.make(binding.root, "Your total amount is $it", Snackbar.LENGTH_LONG).show()
            }
        }
    }
}
```

MainViewModel:

```
class MainViewModel:ViewModel() {
    private val _message = MutableLiveData<String>()
    private val _description = MutableLiveData<String>()
    private val _amount = MutableLiveData<Int>(1) //how many apples
    private val _money = MutableLiveData<Int>() //how much is apple
    private val _qty = MutableLiveData<Int>()
    private val _total = MutableLiveData<Int>()//total of apples

    private var apples: Int = 1
    private var deliveryfee: Int = 45

    val message: LiveData<String> = _message
    val description: LiveData<String> = _description
    val Qty: LiveData<Int> = _qty
    val total: LiveData<Int> = _total

    init {
        _money.value = MainHelper.getApplePrice()
        _description.value = MainHelper.getAppleDesc()
    }

    fun apples(amountType: Int){
        _amount.value = when(amountType){
            1 -> apples++
            2 -> apples--
            else -> 0
        }
        if(apples in 1..10) {
            if(apples > 1){
                _qty.value = apples
                loadMessage()
            }
            else{
                _qty.value = apples
                loadMessage()
            }
        }
    }
    fun totalAmount(){
        if(apples < 10){
            _total.value = _money.value!! * _qty.value!! + deliveryfee
        }
        else{
            _total.value = _money.value!! * _qty.value!!
        }
    }

    fun loadMessage(){
        if(apples < 10){
            _message.value = MainHelper.getMessageFee()
        }
        else{
            _message.value = MainHelper.getMessageFree()
        }
    }
}
```

MainHelper.kt:

```
class MainHelper {
    companion object {
        fun getMessageFee():String{
            return "Delivery Fee: + (Php) 45"
        }
        fun getMessageFree():String{
            return "Delivery Fee: + (Php) 0"
        }
        fun getAppleDesc():String{
            return "An apple is an edible fruit produced by an apple tree (Malus domestica). " +
                "Apple trees are cultivated worldwide and are the most widely grown species in the genus Malus. " +
                "The tree originated in Central Asia, where its wild ancestor, Malus sieversii, is still found today"
        }
        fun getApplePrice():Int{
            return 120
        }
    }
}
```

B. Include a short discussion on how you were able to accomplish the task

For me, what I did to accomplish the challenged that was given to us, I made 6 MutableLiveData that needed for each part of my application and create 2 variables for the apples that will be used for incrementing the quantity and the delivery fee 45Php, and 4 Live data that I will used to pass the data to the main activity.

For the quantity of apples what I did to limit itself to 1 and 10 is to used and if statement that if apples reach in 1 or 10 it will execute another if statement that checks if apples is greater than 1 increment it by 1 or decrement it, else it will remain the same, also made a separate function for the computation that I called out in the main activity, and lastly for the message I made also a function and call it inside the apple increment checker to change the text depending on the quantity

- C. Screen Record a short video demo (average of 30 seconds) of your working app. Create a YouTube channel (if you don't have one yet) and upload your video demo as unlisted. Make sure to include a clickable link of your YouTube video in this report.

[https://youtu.be/pn\\_hifYVq\\_8](https://youtu.be/pn_hifYVq_8)

- D. Self-assessment (what have you learned?).

I have learned that, how to properly use and manipulate MutableLiveData and Live data for the logic of my application, and to make it cleaner in the main activity, also know the importance of using the Main Helper or Separation of Concerns.