

```
1 class Bank {  
2     var ac_no: Int,  
3     var name: String,  
4     var amount: Float,  
5 } {  
6     fun display(){  
7         println("Account Number: ${ac_no}")  
8         println("Name: ${name}")  
9         println("Amount: ${amount}")  
10    }  
11  
12    fun add(a: Float){  
13        println("Amount to be added: $a")  
14        amount += a  
15    }  
16  
17    fun checkBalance(){  
18        println("Balance amount is: $amount")  
19    }  
20  
21    fun remove(b: Float) {  
22        if(amount < b)  
23    }  
24
```

Run:

MainKt ×

```
▶ Account Number: 459231  
↑ Name: Adeel  
↓ Amount: 15000.0  
↔ Amount to be added: 25000.0  
☰ Balance amount is: 40000.0  
☷ Amount to be removed: 12000.0  
☷ Balance amount is: 28000.0
```

```
Main.kt < Bank.kt <
15 }
16
17 fun checkBalance(){
18     println("Balance amount is: $amount")
19 }
20
21 fun remove(b: Float) {
22     if(amount < b)
23         println("Insufficient balance")
24     else
25         amount -= b
26         println("Amount to be removed: $b")
27 }
28
29
30
31
```

Run: MainKt <

Account Number: 459231
Name: Adeel
Amount: 15000.0
Amount to be added: 25000.0
Balance amount is: 40000.0
Amount to be removed: 12000.0
Balance amount is: 28000.0

Main.kt X Bank.kt X

```
> fun main() {
    var b = Bank(ac_no: 459231, name: "Adeel", amount: 15000F)
    b.display()
    b.add(25000F)
    b.checkBalance()
    b.remove(b: 12000F)
    b.checkBalance()
```

Run: MainKt X

```
Account Number: 459231
Name: Adeel
Amount: 15000.0
Amount to be added: 25000.0
Balance amount is: 40000.0
Amount to be removed: 12000.0
Balance amount is: 28000.0
```

Edit View Navigate Code Refactor Build Run Tools VCS Window Help ClassesObjects - ParkingSystem.kt

src > main > kotlin > ParkingSystem > check

ParkingSystem.kt

```
class ParkingSystem {
    var vehicleType: String,
    var entryTime: Int,
    var exitTime: Int,
}
{
    fun display() {
        println("Vehicle Type: ${vehicleType}")
        println("Entry Time: ${entryTime}")
        println("Exit Time: ${exitTime}")
    }
}

fun check() {
    if (vehicleType == "Two Wheeler") {
        var c = exitTime - entryTime
        println("The total amount for parking two wheeler for $c hours is: ${c * 20}")
    } else if (vehicleType == "Four Wheeler") {
        var c = exitTime - entryTime
        println("The total amount for parking four wheeler for $c hours is: ${c * 40}")
    }
}
```

MainKt

Vehicle Type: Four Wheeler
Entry Time: 3
Exit Time: 5
The total amount for parking four wheeler for 2 hours is: 80

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Classes Objects - Main.kt

Objects > src > main > kotlin > Main.kt

Main.kt × ParkingSystem.kt ×

```
fun main() {
    var amt = ParkingSystem( vehicleType: "Four Wheeler", entryTime: 3, exitTime: 5)
    amt.display()
    amt.check()
```

I

MainKt ×

```
Vehicle Type: Four Wheeler
Entry Time: 3
Exit Time: 5
The total amount for parking four wheeler for 2 hours is: 80
```

```
1 class TransactionSystem {
2     var amount: Int,
3     var amt_added: Int,
4     var reasonForAdd: String,
5     var amt_spent: Int,
6     var reasonForSpent: String,
7 }
8 fun display() {
9     var balanceAdd = amount + amt_added
10    var balanceRemove = balanceAdd - amt_spent
11    println("${amt_added}\t${reasonForAdd}\t${balanceAdd}\t${amt_spent}\t${reasonForSpent}\t${balanceRemove}")
12 }
13 }
```

Run: MainKt

"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition

AMOUNT					
ADDED	REASON	Balance	SPENT	REASON	Balance
5000	Savings	30000	10000	Training	20000
2000	Salary	47000	6500	Bills	40500
9000	Bonus	41000	3500	Essentials	37500

Process finished with exit code 0

Run TODO Problems Terminal Build

```
1 > fun main(){
2     println("                AMOUNT")
3     println("ADDED\tREASON\tBalance\tSPENT\t REASON\t Balance")
4     var t = TransactionSystem( amount: 25000, amt_added: 5000, reasonForAdd: "Savings", amt_spent: 10000, reasonForSpent: "Training" )
5     t.display()
6     var p = TransactionSystem( amount: 45000, amt_added: 2000, reasonForAdd: "Salary", amt_spent: 6500, reasonForSpent: "Bills" )
7     p.display()
8     var r = TransactionSystem( amount: 32000, amt_added: 9000, reasonForAdd: "Bonus", amt_spent: 3500, reasonForSpent: "Essentials" )
9     r.display()
10 }
```

I

Run: MainKt ×

```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.1\lib\idea_rt.jar=5314,C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.1\bin" -Dfile.encoding=UTF-8
                AMOUNT
    ADDED   REASON   Balance SPENT      REASON      Balance
    5000    Savings  30000   10000    Training    20000
    2000    Salary   47000   6500     Bills       40500
    9000    Bonus    41000   3500     Essentials  37500
```

Process finished with exit code 0

```
class Teachers{  
    private val specialization: String  
) : User( name: "Rita", department: "Computer Science", id_no: 3462){  
  
    init{  
        println("Teachers details")  
        println("Name: $name")  
        println("Department: $department")  
        println("ID Number: $id_no")  
        println("Specialization: $specialization")  
    }  
}
```

MainKt

Name: Arya
Department: Electronics and Communication
ID Number: 2737
Section: B

Teachers details
Name: Rita
Department: Computer Science
ID Number: 3462
Specialization: Cloud Computing

```
class Students(  
    val section: Char  
) : User( name: "Arya", department: "Electronics and Communication",  
  
    init{  
        println("Student details")  
        println("Name: $name")  
        println("Department: $department")  
        println("ID Number: $id_no")  
        println("Section: $section")  
        println()  
    }  
}
```

MainKt ×

Name: Arya
Department: Electronics and Communication
ID Number: 2737
Section: B

Teachers details
Name: Rita
Department: Computer Science
ID Number: 3462
Specialization: Cloud Computing

ain.kt × User.kt × Students.kt × Teachers.kt ×

```
open class User(  
    var name: String,  
    var department: String,  
    var id_no: Int  
)
```

MainKt ×

Name: Arya

Department: Electronics and Communication

ID Number: 2737

Section: B

Teachers details

Name: Rita

Department: Computer Science

ID Number: 3462

Specialization: Cloud Computing

```
> fun main(){  
    val m = Students( section: 'B')  
    val n = Teachers( specialization: "Cloud Computing")  
}
```

I

MainKt

Name: Arya
Department: Electronics and Communication
ID Number: 2737
Section: B

Teachers details
Name: Rita
Department: Computer Science
ID Number: 3462
Specialization: Cloud Computing