
Mobile App Development Lab Assignment 1 - Execute Kotlin Programs

From:

Zaheer Abbas

4JN18CS128

6th sem CSE, B

JNNCE

To:

Dr. Chetan K.R

Associate Professor

Computer Science Department

JNNCE

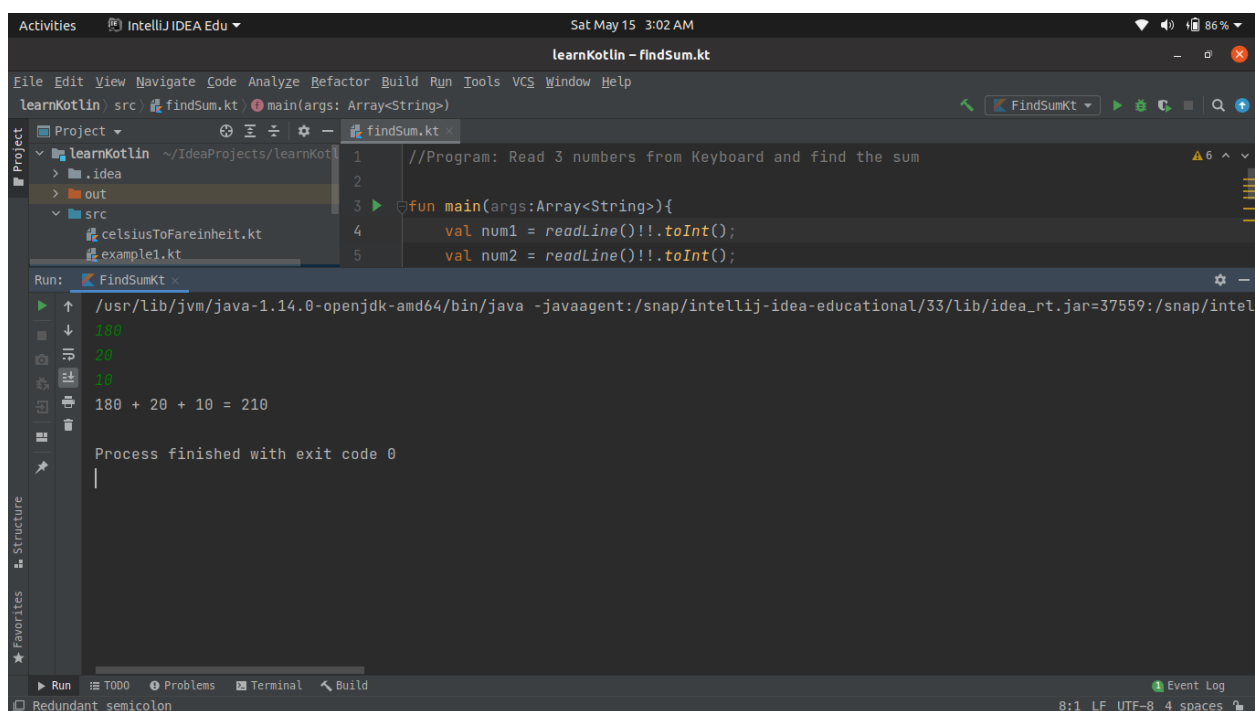
All the code can be found [here](#):

(Link url: <https://github.com/nk4456542/learnKotlin/tree/main/src>)

1. Program: Read 3 numbers from Keyboard and find the sum

```
//Program: Read 3 numbers from Keyboard and find the sum
fun main(args:Array<String>){
    val num1 = readLine()!!.toInt();
    val num2 = readLine()!!.toInt();
    val num3 = readLine()!!.toInt();
    val sum = num1 + num2 + num3;
    println("$num1 + $num2 + $num3 = $sum");
}
```

Output:



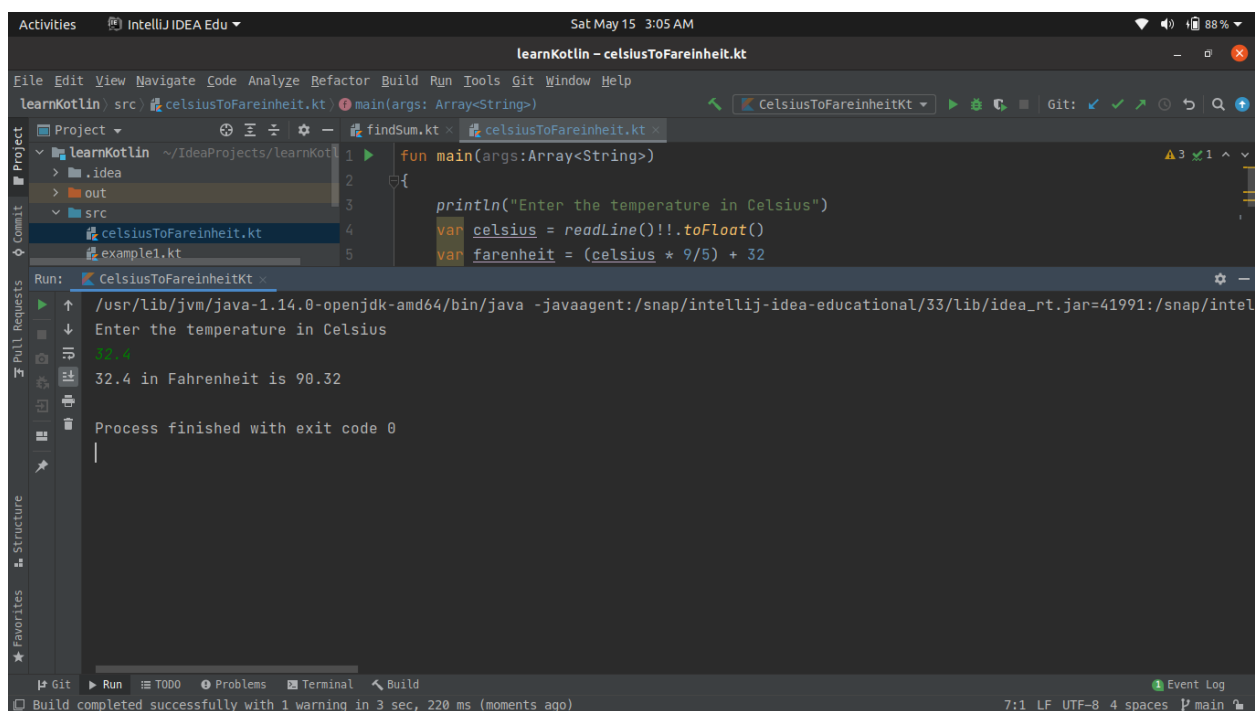
The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The main editor window shows the Kotlin code for the program. Below the editor, the 'Run' console is visible, showing the output of the program. The output indicates that the program executed successfully, reading three numbers (180, 20, 10) and printing their sum (210). The console also shows the command used to run the program: `/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=37559:/snap/intel`. The status bar at the bottom indicates the file encoding is UTF-8 and the line length is 8:1.

```
Run: FindSumKt x
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=37559:/snap/intel
180
20
10
180 + 20 + 10 = 210
Process finished with exit code 0
```

2. Program: Read temperature in Celsius and convert to Fahrenheit

```
//Program: Read temperature in Celsius and convert to Fahrenheit  
fun main(args:Array<String>)  
{  
    println("Enter the temperature in Celsius")  
    var celsius = readLine()!!.toFloat()  
    var fahrenheit = (celsius * 9/5) + 32  
    println("$celsius in Fahrenheit is $fahrenheit")  
}
```

Output:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The main editor window shows the Kotlin code for the temperature conversion program. Below the code editor, the 'Run' console is visible, showing the execution output. The output indicates that the program was run successfully, with the input '32.4' and the output '32.4 in Fahrenheit is 90.32'. The status bar at the bottom shows 'Build completed successfully with 1 warning in 3 sec, 220 ms (moments ago)'.

```
Run: CelsiusToFahrenheitKt  
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=41991:/snap/intel  
Enter the temperature in Celsius  
32.4  
32.4 in Fahrenheit is 90.32  
Process finished with exit code 0
```

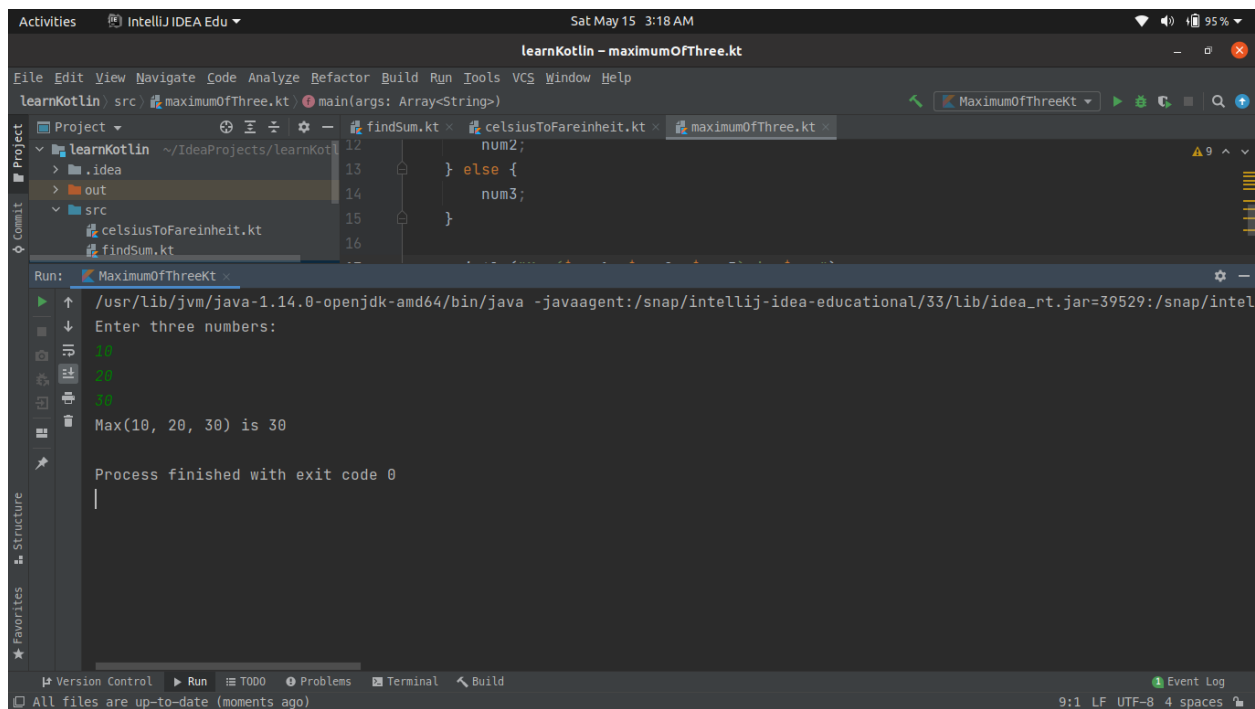
Build completed successfully with 1 warning in 3 sec, 220 ms (moments ago)

3. Program: Find maximum of 3 numbers

//Program: Find maximum of 3 numbers

```
fun main(args:Array<String>){
    println("Enter three numbers:");
    val num1:Int = readLine()!!.toInt();
    val num2:Int = readLine()!!.toInt();
    val num3:Int = readLine()!!.toInt();
    val max = if(num1 > num2 && num1 > num3){
        num1;
    } else if(num2 > num1 && num2 > num3){
        num2;
    } else {
        num3;
    }
    println("Max($num1, $num2, $num3) is $max");
}
```

Output:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The main editor window is titled 'learnKotlin - maximumOfThree.kt' and shows the following Kotlin code:

```
12     num2;
13     } else {
14         num3;
15     }
16 }
```

Below the editor, the 'Run' console is visible, showing the output of the program:

```
Run: MaximumOfThreeKt
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=39529:/snap/intel
Enter three numbers:
10
20
30
Max(10, 20, 30) is 30
Process finished with exit code 0
```

The status bar at the bottom indicates 'All files are up-to-date (moments ago)' and '9:1 LF UTF-8 4 spaces'.

4. Program: Print class based on result percentage

```
//Program: Print class based on result percentage
fun addClassToResult(result:Double):String{
    val classBasedOnResult = when (result) {
        in 85.00..100.00 -> "Distinction Class"
        in 65.00..84.99 -> "First Class"
        in 35.00..64.99 -> "Second Class"
        else -> "Fail Class"
    }
    return classBasedOnResult;
}

fun main(args:Array<String>) {
    println("Enter your result in percentage:")
    val result:Double = readLine()!!.toDouble();
    val classBasedOnResult = addClassToResult(result);
    println("You belong to $classBasedOnResult based on your
result: $result");

    println("Some sample results")
    val resultArray = doubleArrayOf(14.91,45.0,78.9,89.1,95.1);
    for(resultItem in resultArray) {
        val classResult = addClassToResult(resultItem)
        println("You belong to $classResult based on your
result: $resultItem");
    }
}
```

Output:

```
learnKotlin - percentageClass.kt
14 val classBasedOnResult = addClassToResult(result);
15 println("You belong to $classBasedOnResult based on your result: $result");
16 println("Some sample results")
17 val resultArray = doubleArrayOf(14.91, 45.0, 78.9, 89.1, 95.1);
18 for(resultItem in resultArray) {
```

Run: PercentageClassKt

```
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=36459:/snap/intel
Enter your result in percentage:
95.6
You belong to Distinction Class based on your result: 95.6
Some sample results
You belong to Fail Class based on your result: 14.91
You belong to Second Class based on your result: 45.0
You belong to First Class based on your result: 78.9
You belong to Distinction Class based on your result: 89.1
You belong to Distinction Class based on your result: 95.1

Process finished with exit code 0
```

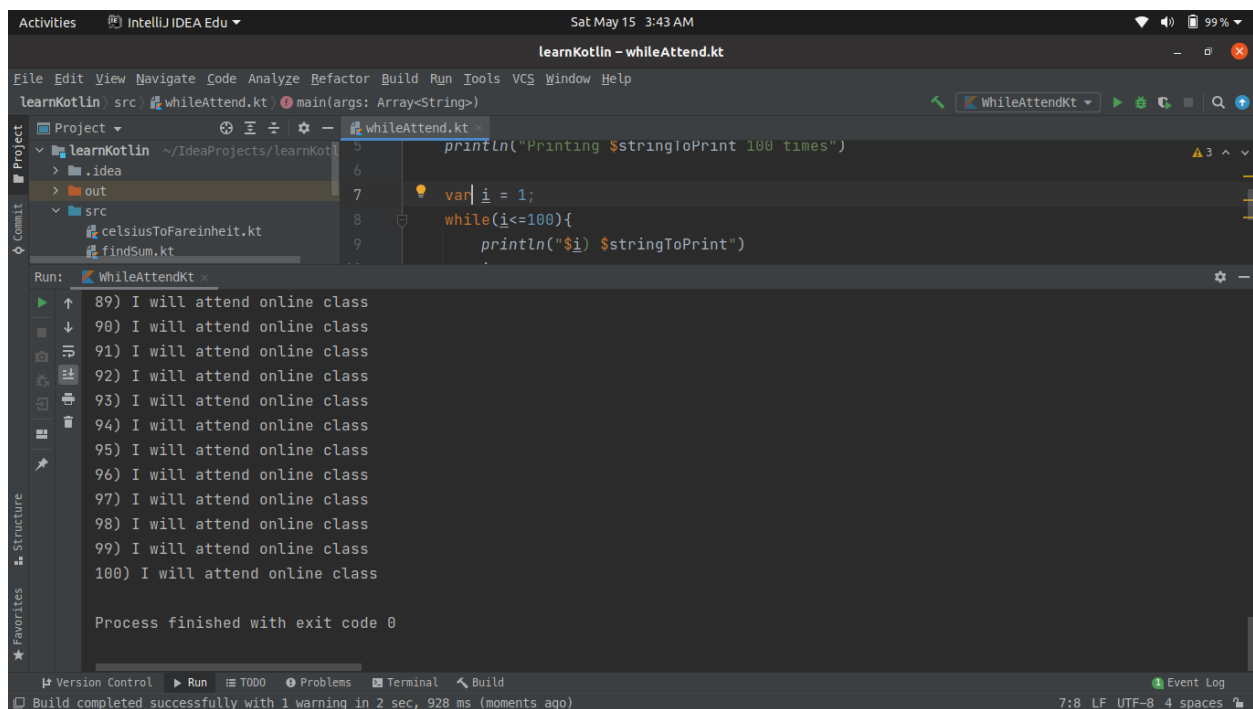
Build completed successfully with 1 warning in 2 sec, 886 ms (moments ago)

5. Program: Print “I will attend online class” 100 times

```
//Program: Print "I will attend online class" 100 times
fun main(args:Array<String>){
    val stringToPrint = "I will attend online class"
    println("Printing $stringToPrint 100 times")

    var i = 1;
    while(i<=100){
        println("$i) $stringToPrint")
        i++;
    }
}
```

Output:

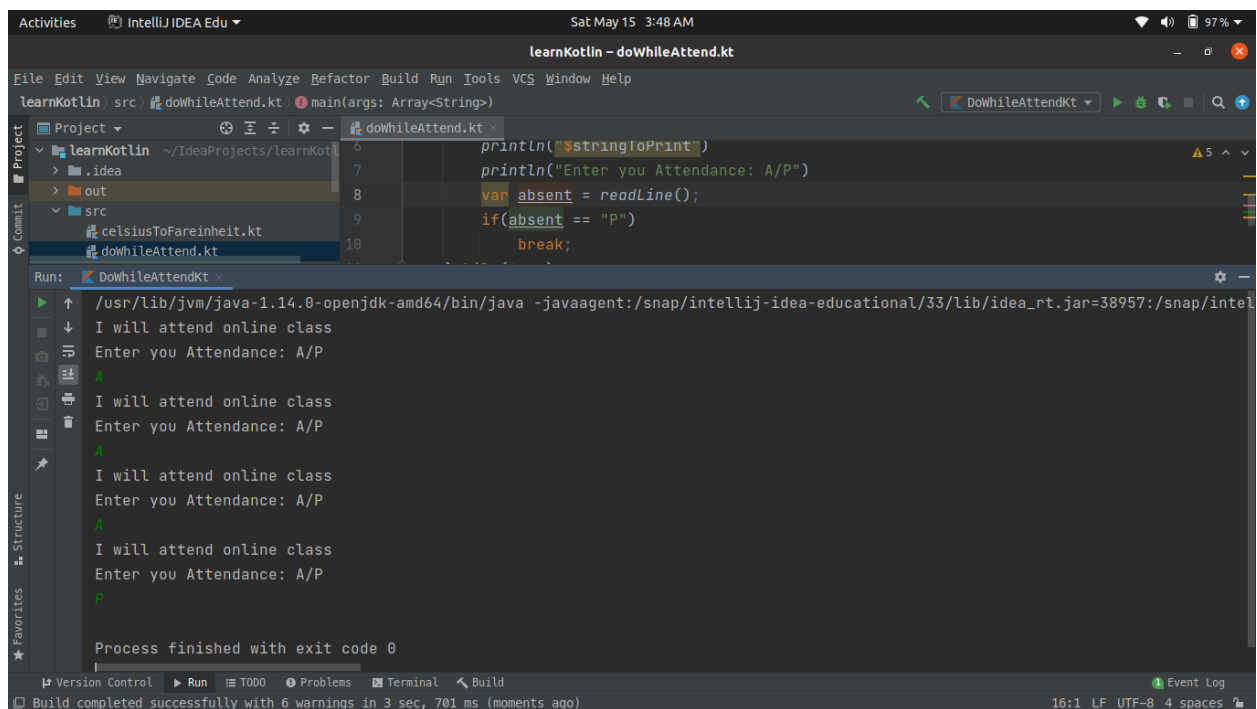


```
Activities IntelliJ IDEA Edu Sat May 15 3:43 AM
learnKotlin - whileAttend.kt
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
learnKotlin src whileAttend.kt main(args: Array<String>)
Project learnKotlin ~/IdeaProjects/learnKot
> .idea
> out
src
celsiusToFahrenheit.kt
findSum.kt
whileAttend.kt
5 println("Printing $stringToPrint 100 times")
6
7 var i = 1;
8 while(i<=100){
9     println("$i) $stringToPrint")
10 }
Run: WhileAttendKt x
89) I will attend online class
90) I will attend online class
91) I will attend online class
92) I will attend online class
93) I will attend online class
94) I will attend online class
95) I will attend online class
96) I will attend online class
97) I will attend online class
98) I will attend online class
99) I will attend online class
100) I will attend online class
Process finished with exit code 0
Build completed successfully with 1 warning in 2 sec, 928 ms (moments ago) 7:8 LF UTF-8 4 spaces
```

6. Program: Print “I will attend online class” until you attend

```
//Program: Print "I will attend online class" until you attend
fun main(args:Array<String>){
    val stringToPrint = "I will attend online class"
    do {
        println("$stringToPrint")
        println("Enter you Attendance: A/P")
        var absent = readLine();
        if(absent == "P")
            break;
    }while(true)
}
```

Output:



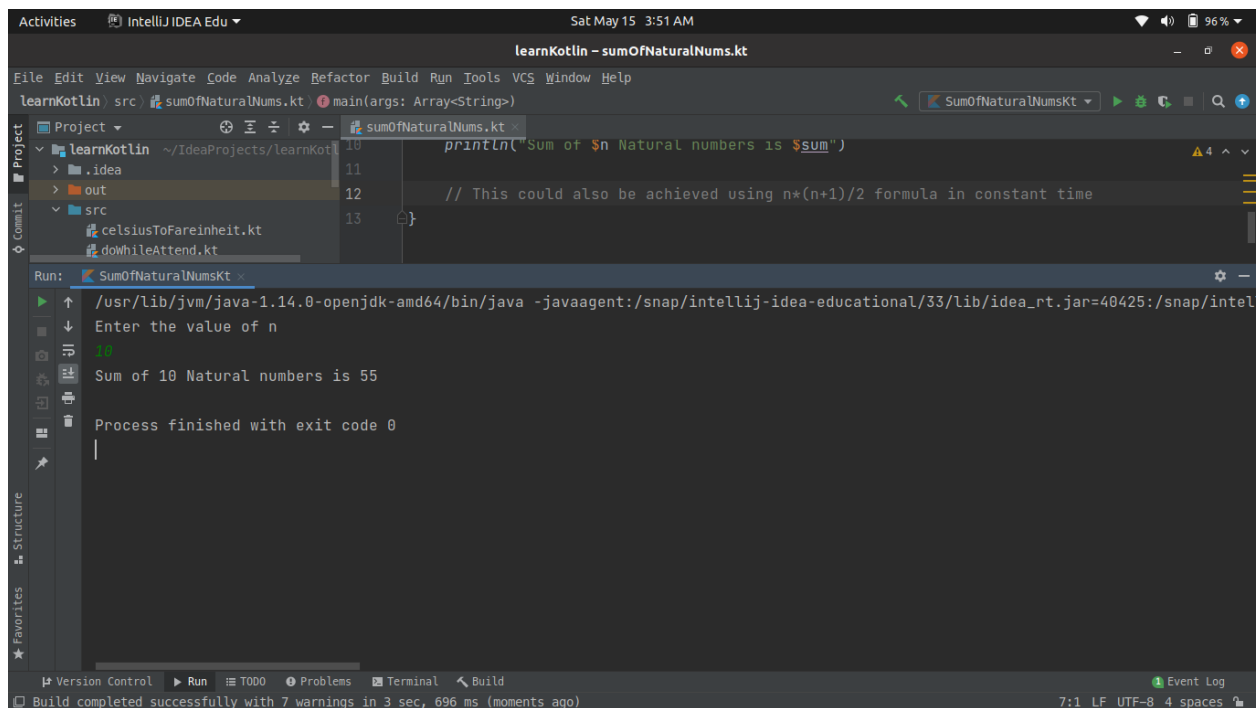
```
Run: DowhileAttendKt x
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=38957:/snap/intel
I will attend online class
Enter you Attendance: A/P
I will attend online class
Enter you Attendance: A/P
I will attend online class
Enter you Attendance: A/P
I will attend online class
Enter you Attendance: A/P
Process finished with exit code 0
```


7. Program: Find sum of n natural numbers

```
//Program: Find sum of n natural numbers
fun main(args:Array<String>){
    println("Enter the value of n");
    val n:Int = readLine()!!.toInt()
    var sum = 0;
    for(i in 1..n){
        sum += i;
    }
    println("Sum of $n Natural numbers is $sum")

    // This could also be achieved using  $n*(n+1)/2$  formula in
    constant time
}
```

Output:



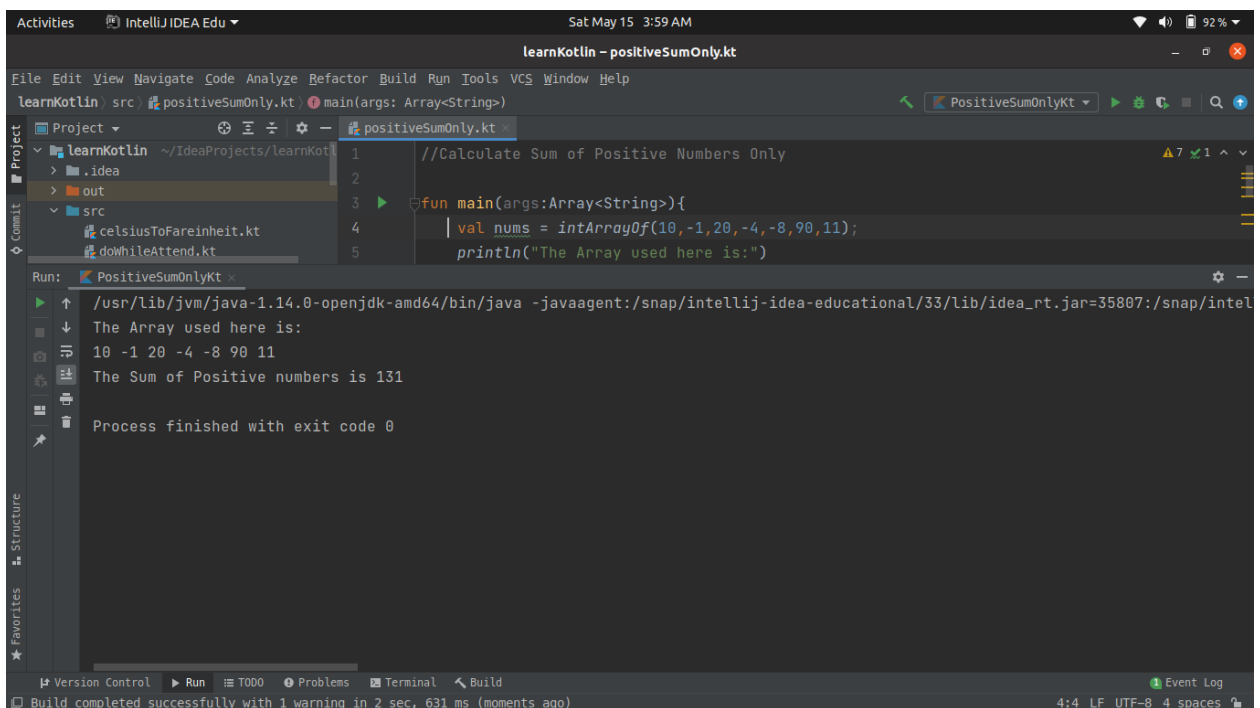
The screenshot shows the IntelliJ IDEA IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The main editor displays the Kotlin code from the previous block. The Run console at the bottom shows the execution output: "Enter the value of n", "10", "Sum of 10 Natural numbers is 55", and "Process finished with exit code 0". The status bar at the bottom indicates "Build completed successfully with 7 warnings in 3 sec, 696 ms (moments ago)" and "7:1 LF UTF-8 4 spaces".

```
Run: SumOfNaturalNumsKt
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=40425:/snap/intel
Enter the value of n
10
Sum of 10 Natural numbers is 55
Process finished with exit code 0
```

8. Example: Calculate Sum of Positive Numbers Only

```
//Calculate Sum of Positive Numbers Only
fun main(args:Array<String>){
    val nums = intArrayOf(10,-1,20,-4,-8,90,11);
    println("The Array used here is:")
    var sum:Int = 0;
    for(num in nums) {
        print("$num ");
        if (num <= 0) {
            continue
        }
        sum += num;
    }
    println("\nThe Sum of Positive numbers is $sum");
}
```

Output:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The main editor window shows the Kotlin code for 'positiveSumOnly.kt'. Below the editor, the 'Run' console is visible, showing the output of the program. The output is as follows:

```
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=35807:/snap/intel
The Array used here is:
10 -1 20 -4 -8 90 11
The Sum of Positive numbers is 131
Process finished with exit code 0
```

The status bar at the bottom indicates 'Build completed successfully with 1 warning in 2 sec, 631 ms (moments ago)'.

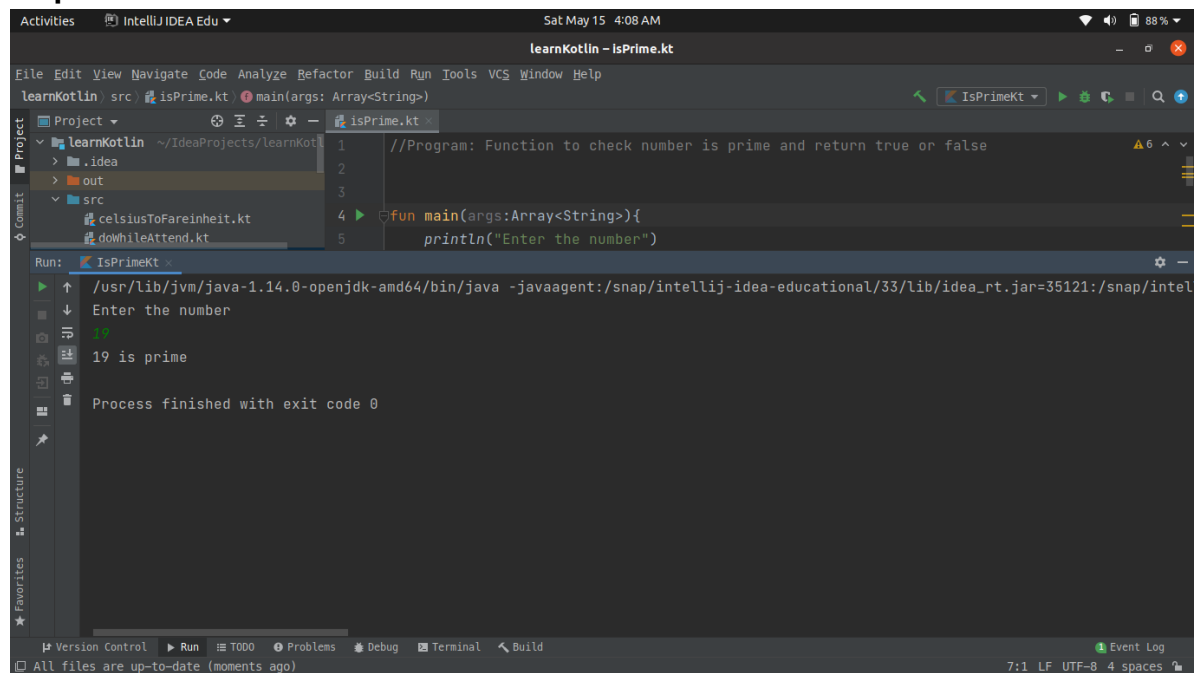
9. Program: Function to check number is prime and return true or false

//Program: Function to check number is prime and return true or false

```
fun isPrime(num:Int):Boolean{
    for(i in 2..num/2){
        if(num%i == 0){
            return false;
        }
    }
    return true;
}

fun main(args:Array<String>){
    println("Enter the number")
    val num:Int = readLine()!!.toInt();
    var result:Boolean = isPrime(num);
    if(result) {
        println("$num is prime")
    }else {
        println("$num is not prime")
    }
}
```

Output:



The screenshot displays the IntelliJ IDEA IDE interface. The top bar shows 'Sat May 15 4:08 AM' and '88%' battery. The title bar reads 'learnKotlin - IsPrime.kt'. The menu bar includes 'File', 'Edit', 'View', 'Navigate', 'Code', 'Analyze', 'Refactor', 'Build', 'Run', 'Tools', 'VCS', 'Window', and 'Help'. The toolbar shows icons for running and debugging. The 'Project' view on the left shows the project structure: 'learnKotlin' (root), '.idea', 'out', 'src' (containing 'celsiusToFahrenheit.kt' and 'dowhileAttend.kt'), and 'IsPrime.kt' (selected). The editor window shows the following Kotlin code:

```
1 //Program: Function to check number is prime and return true or false
2
3
4 fun main(args:Array<String>){
5     println("Enter the number")
```

The 'Run' view at the bottom shows the execution output for 'IsPrimeKt':

```
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=35121:/snap/intel
Enter the number
19
19 is prime
Process finished with exit code 0
```

The status bar at the bottom indicates 'All files are up-to-date (moments ago)' and '7:1 LF UTF-8 4 spaces'.

10. Program to demonstrate primary constructors

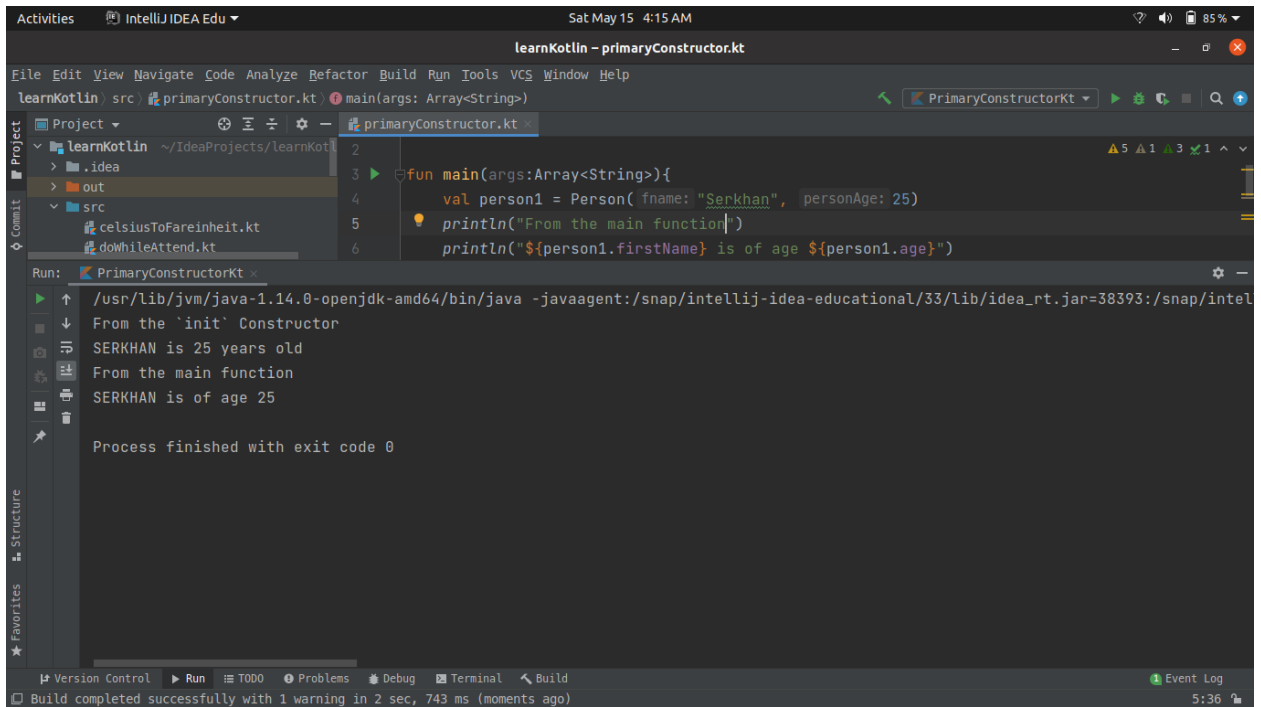
//Program to demonstrate primary constructors

```
fun main(args:Array<String>){
    val person1 = Person("Serkhan", 25)
    println("From the main function")
    println("${person1.firstName} is of age ${person1.age}")
}

class Person(val fname:String, var personAge:Int){
    val firstName:String
    val age:Int

    init {
        firstName = fname.uppercase();
        age = personAge;
        println("From the `init` Constructor")
        println("$firstName is $age years old")
    }
}
```

Output:



The screenshot displays the IntelliJ IDEA Edu interface. The top bar shows the title 'learnKotlin - primaryConstructor.kt' and the date 'Sat May 15 4:15 AM'. The main editor window shows the following Kotlin code:

```
1 // ...  
2  
3 fun main(args:Array<String>){  
4     val person1 = Person( fname: "Serkan", personAge: 25)  
5     println("From the main function")  
6     println("${person1.firstName} is of age ${person1.age}")  
}
```

The left sidebar shows the project structure with 'learnKotlin' as the root, containing '.idea', 'out', and 'src' directories. The 'src' directory contains 'celsiusToFahrenheit.kt' and 'dowhileAttend.kt'. The bottom panel shows the 'Run' output for 'PrimaryConstructorKt':

```
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=38393:/snap/intel  
From the 'init' Constructor  
SERKHAN is 25 years old  
From the main function  
SERKHAN is of age 25  
  
Process finished with exit code 0
```

The bottom status bar indicates 'Build completed successfully with 1 warning in 2 sec, 743 ms (moments ago)' and the time '5:36'.

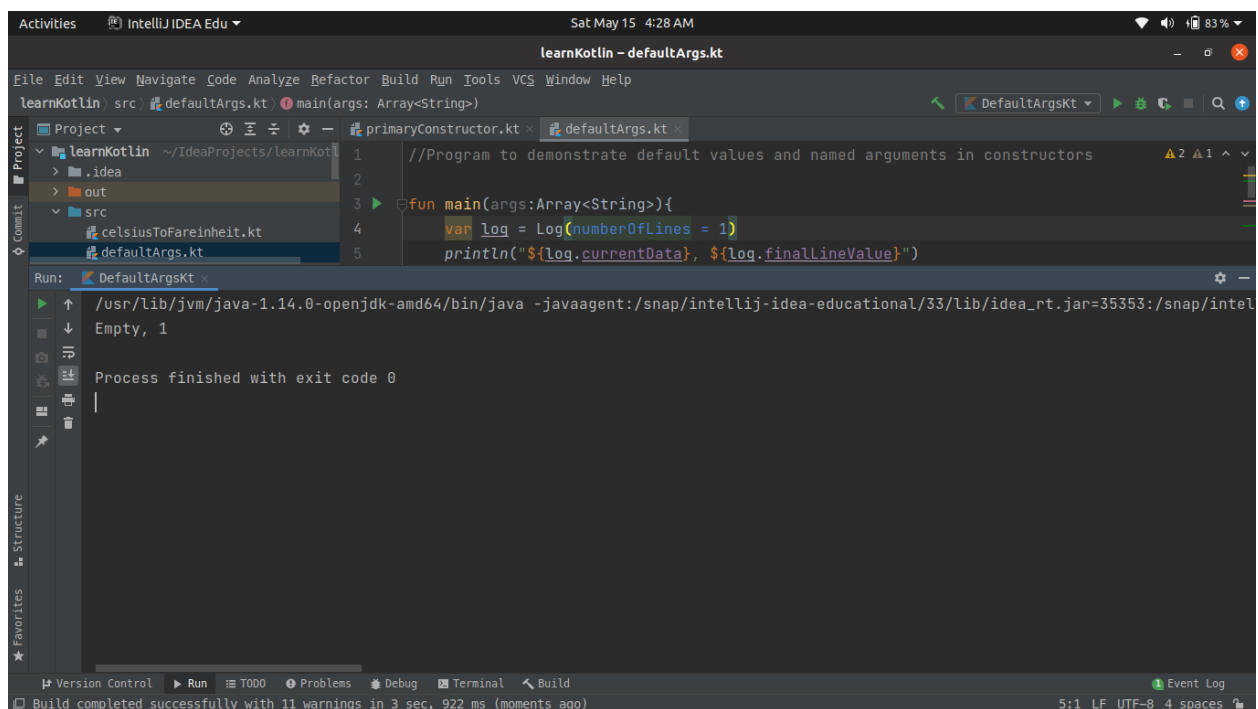
11. Program to demonstrate default values and named arguments in constructors

//Program to demonstrate default values and named arguments in constructors

```
fun main(args:Array<String>){
    var log = Log(numberOfLines = 1)
    println("${log.currentData}, ${log.finallineValue}")
}

class Log(data:String="Empty",numberOfLines:Int=0){
    var currentData:String
    var finalLineValue:Int
    init {
        currentData = data
        finalLineValue = numberOfLines
    }
}
```

Output:



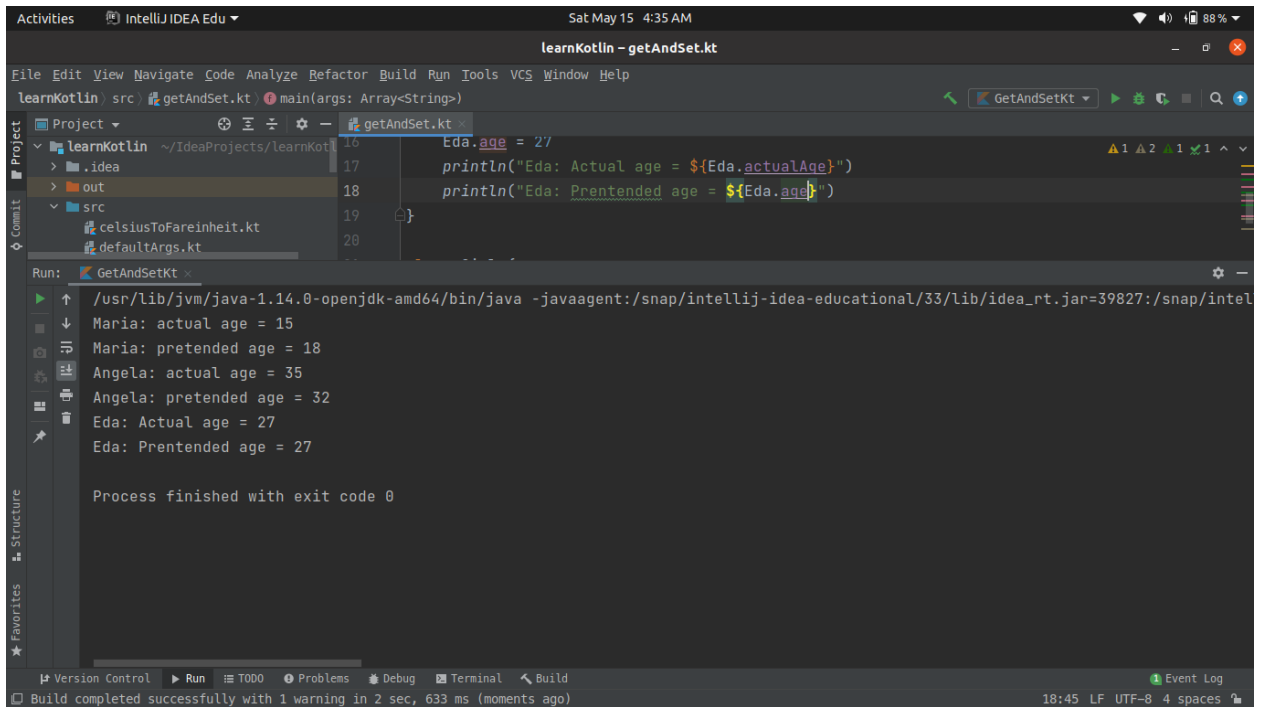
12. Program to demonstrate the use of getters and setters

//Program to demonstrate the use of getters and setters

```
fun main(args: Array<String>) {
    val maria = Girl()
    maria.actualAge = 15
    maria.age = 15
    println("Maria: actual age = ${maria.actualAge}")
    println("Maria: pretended age = ${maria.age}")
    val angela = Girl()
    angela.actualAge = 35
    angela.age = 35
    println("Angela: actual age = ${angela.actualAge}")
    println("Angela: pretended age = ${angela.age}")
    val Eda = Girl()
    Eda.actualAge = 27
    Eda.age = 27
    println("Eda: Actual age = ${Eda.actualAge}")
    println("Eda: Pretended age = ${Eda.age}")
}

class Girl {
    var age: Int = 0
    get() = field
    set(value) {
        field = if (value < 18)
            18
        else if (value >= 18 && value <= 30)
            value
        else
            value - 3
    }
    var actualAge: Int = 0
}
```


Output:



The screenshot shows the IntelliJ IDEA IDE interface. The top bar indicates the project is 'learnKotlin - getAndSet.kt' and the date is 'Sat May 15 4:35 AM'. The main editor displays the following Kotlin code:

```
16 Eda.age = 27
17 println("Eda: Actual age = ${Eda.actualAge}")
18 println("Eda: Pretended age = ${Eda.age}")
19 }
20
```

The 'Run' tab at the bottom shows the execution output:

```
/usr/lib/jvm/java-1.14.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-educational/33/lib/idea_rt.jar=39827:/snap/intel
Maria: actual age = 15
Maria: pretended age = 18
Angela: actual age = 35
Angela: pretended age = 32
Eda: Actual age = 27
Eda: Pretended age = 27

Process finished with exit code 0
```

The status bar at the bottom indicates 'Build completed successfully with 1 warning in 2 sec, 633 ms (moments ago)' and the time is '18:45'.

13. Program to illustrate the use of Nested Inner class

//Program to illustrate the use of Nested Inner class

```
class Outer {  
    val a = "Outside Nested class."  
    inner class Inner {  
        fun callMe() = a  
    }  
}  
  
fun main(args: Array<String>) {  
    val outer = Outer()  
    println("Using outer object: ${outer.Inner().callMe()}")  
    val inner = Outer().Inner()  
    println("Using inner object: ${inner.callMe()}")  
}
```

Output:

