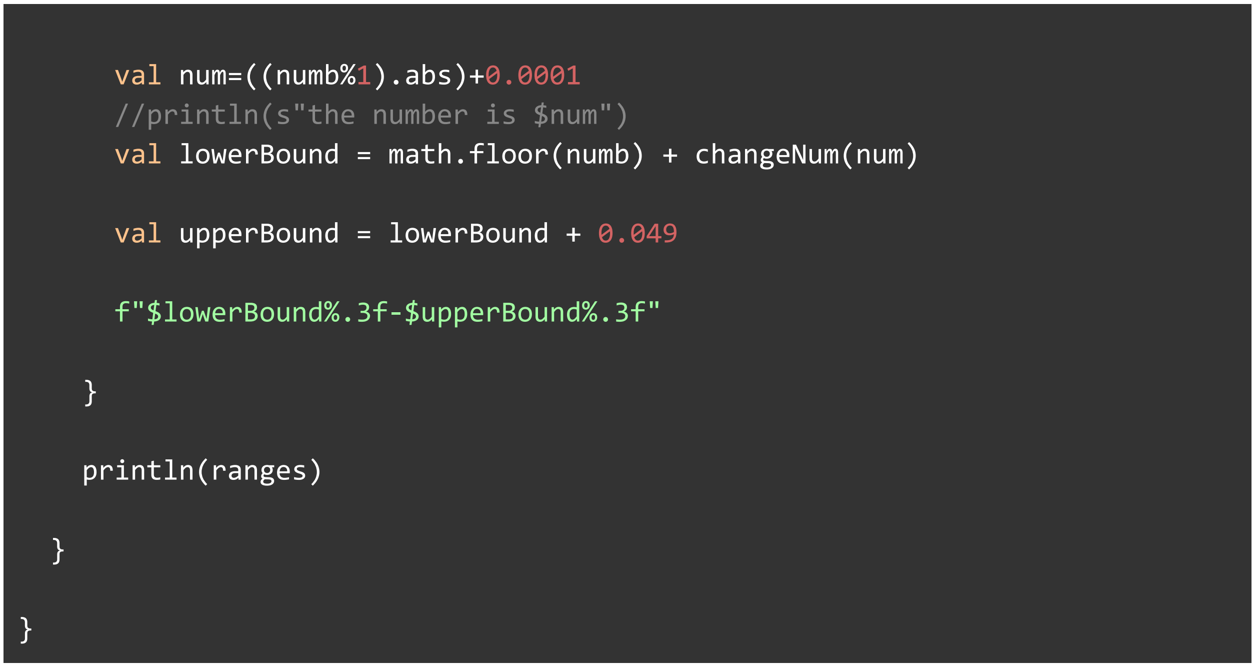
**Name- Rajkeshav Kumar Jha**
**TAS129**

# Scala Assignment

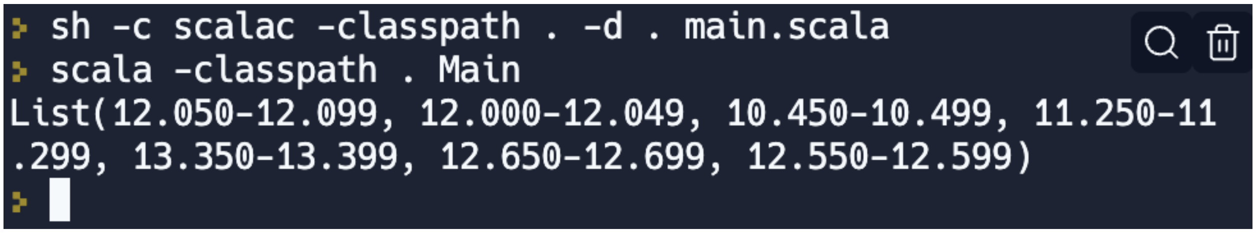
**Q1)**

```
object Main{
  private def changeNum(d: Double):Double={
    val temp=d*1000
//    println(s"temp value ="+temp)
    val remByHun=temp%100
//    println(remByHun)
    var desiredOutput=0.000
    if(remByHun < 50)
      desiredOutput=(temp-remByHun)/1000.000
    else
      desiredOutput=(temp-remByHun)/1000.00 +0.050
    desiredOutput
  }
  def main(args:Array[String])={
    val numbers = List(12.05, 12.03, 10.45, 11.299, 13.350,12.677,12.55)
    val ranges = numbers.map { numb =>
//      println(numb)
```



```
      val num=((numb%1).abs)+0.0001
      //println(s"the number is $num")
      val lowerBound = math.floor(numb) + changeNum(num)

      val upperBound = lowerBound + 0.049

      f"$lowerBound%.3f-$upperBound%.3f"

    }

    println(ranges)

  }

}
```

**Output:**

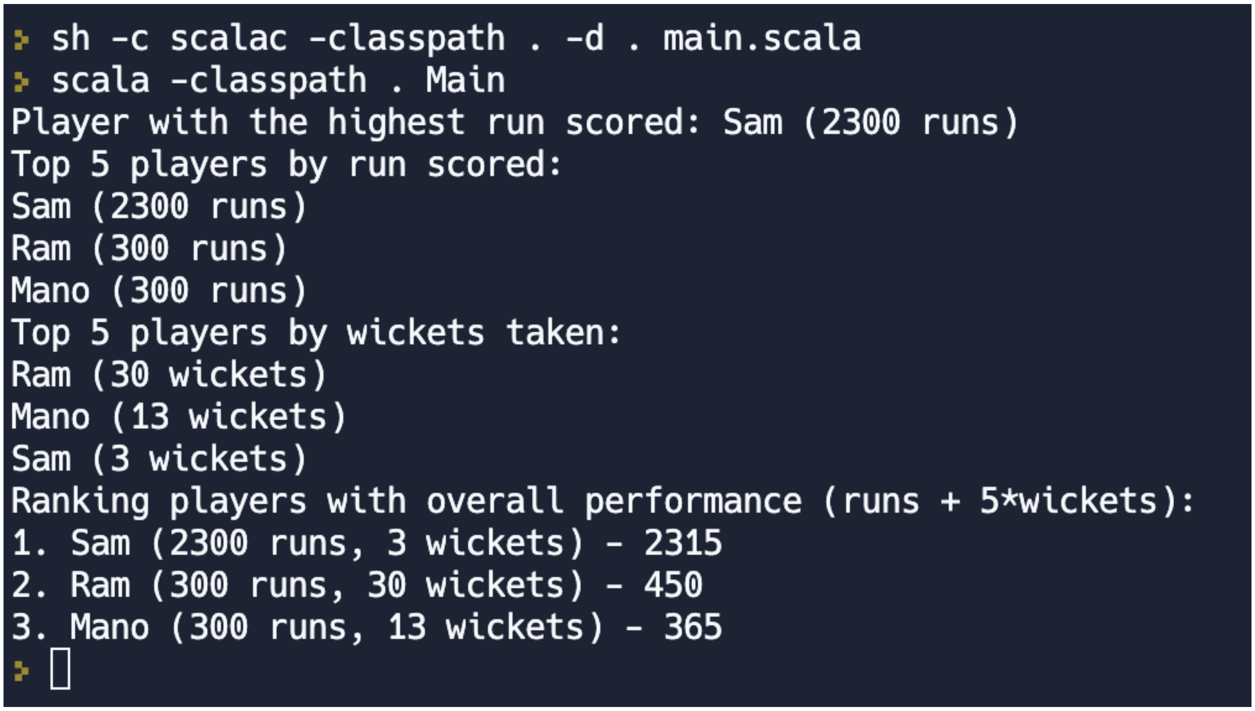


```
> sh -c scalac -classpath . -d . main.scala
> scala -classpath . Main
List(12.050-12.099, 12.000-12.049, 10.450-10.499, 11.250-11
.299, 13.350-13.399, 12.650-12.699, 12.550-12.599)
> 
```

**Q2)**

```
object PlayerStats {
  case class Player(year: Int, name: String, country: String, matches: Int,
runs: Int, wickets: Int)

  def main(args: Array[String]): Unit = {
    // Sample data
    val players = Seq(
      Player(2021, "Sam", "India", 23, 2300, 3),
      Player(2021, "Ram", "India", 23, 300, 30),
      Player(2021, "Mano", "India", 23, 300, 13)
    )

    // 1. Player with the highest run scored
    val playerWithHighestRun = players.maxBy(_.runs)
    println(s"Player with the highest run scored:
${playerWithHighestRun.name} (${playerWithHighestRun.runs} runs)")



 // 2. Top 5 players by run scored

val top5PlayersByRun =
players.sortBy(_.runs)(Ordering[Int].reverse).take(5)
    println("Top 5 players by run scored:")
    top5PlayersByRun.foreach(player => println(s"${player.name}
(${player.runs} runs)"))

    // 3. Top 5 players by wicket taken
    val top5PlayersByWickets =
players.sortBy(_.wickets)(Ordering[Int].reverse).take(5)
    println("Top 5 players by wickets taken:")
    top5PlayersByWickets.foreach(player => println(s"${player.name}
(${player.wickets} wickets)"))

    // 4. Rank players with overall performance
    val rankedPlayers = players.map { player =>
      val overallPerformance = player.runs + (player.wickets * 5)
      (player, overallPerformance)
    }.sortBy(_._2)(Ordering[Int].reverse)
    println("Ranking players with overall performance (runs + 5*wickets):")
    rankedPlayers.zipWithIndex.foreach { case ((player, score), index) =>
      println(s"${index + 1}. ${player.name} (${player.runs} runs,
${player.wickets} wickets) - $score")
    }
  }
}
```

**Output:**



```
› sh -c scalac -classpath . -d . main.scala
› scala -classpath . Main
Player with the highest run scored: Sam (2300 runs)
Top 5 players by run scored:
Sam (2300 runs)
Ram (300 runs)
Mano (300 runs)
Top 5 players by wickets taken:
Ram (30 wickets)
Mano (13 wickets)
Sam (3 wickets)
Ranking players with overall performance (runs + 5*wickets):
1. Sam (2300 runs, 3 wickets) - 2315
2. Ram (300 runs, 30 wickets) - 450
3. Mano (300 runs, 13 wickets) - 365
› 
```