

22000862

R.K.J.P.Kashmira

Q1

The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'SCALA-PRACTICAL-10'. The 'EXPLORER' pane shows files like .bsp, .metals, .scala-build, .vscode, .gitattributes, .gitignore, Q1.scala, Q2.scala, Q3.scala, Q4.scala, Q5.scala, README.md, and tempCodeRunnerFile.scala. The 'Q1.scala' file is open in the editor, showing the following Scala code:

```
1 class Rational(n: Int, d: Int) {  
2   require(d != 0, "Denominator must be non-zero")  
3  
4   private def gcd(a: Int, b: Int): Int = if (b == 0) a else gcd(b, a % b)  
5   private val g = gcd(n.abs, d.abs)  
6  
7   val numer = n / g  
8   val denom = d / g  
9  
10  def neg: Rational = new Rational(-numer, denom)  
11  override def toString: String = numer + "/" + denom  
12  
13  
14  object Main {  
15    def main(args: Array[String]): Unit = {  
16      val x = new Rational(1, 3)  
17      val negX = x.neg  
18      println(negX)  
19    }  
20  }  
21
```

The 'TERMINAL' pane at the bottom shows the command prompt output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10\Q1.scala"  
there was 1 deprecation warning; re-run with -deprecation for details  
1 warning found  
-1/3  
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10>
```

Q2

The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'SCALA-PRACTICAL-10'. The 'EXPLORER' pane shows files like .bsp, .metals, .scala-build, .vscode, .gitattributes, .gitignore, Q1.scala, Q2.scala, Q3.scala, Q4.scala, Q5.scala, README.md, and tempCodeRunnerFile.scala. The 'Q2.scala' file is open in the editor, showing the following Scala code:

```
1 class Rational(n: Int, d: Int) {  
2   require(d != 0, "Denominator must be non-zero")  
3  
4   private val g = gcd(n.abs, d.abs)  
5   val numer: Int = n / g  
6   val denom: Int = d / g  
7  
8   def neg: Rational = new Rational(-numer, denom)  
9  
10  def sub(that: Rational): Rational = {  
11    new Rational(  
12      numer * that.denom - that.numer * denom,  
13      denom * that.denom  
14    )  
15  }  
16  
17  private def gcd(a: Int, b: Int): Int = {  
18    if (b == 0) a else gcd(b, a % b)  
19  }  
20  
21  override def toString: String = s"$numer/$denom"  
22
```

The 'TERMINAL' pane at the bottom shows the command prompt output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10\Q2.scala"  
x: 3/4  
y: 5/8  
z: 2/7  
y - z: 19/56  
x - (y - z): 23/56  
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10\Q2.scala"  
x: 3/4  
y: 5/8  
z: 2/7  
y - z: 19/56  
x - (y - z): 23/56  
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10>
```

Q3

The screenshot shows the VS Code editor with a Scala file named Q3.scala. The code defines an Account class with deposit and withdraw methods. The terminal output shows the execution of the code, demonstrating the deposit and withdraw functionality.

```
class Account(private var balance: Double) {  
  def deposit(amount: Double): Unit = {  
    if (amount > 0) {  
      balance += amount  
      println(s"Deposited ${amount}. New balance: ${balance}")  
    } else {  
      println("Deposit amount must be positive.")  
    }  
  }  
  def withdraw(amount: Double): Unit = {  
    if (amount > 0 && amount <= balance) {  
      balance -= amount  
      println(s"Withdrew ${amount} LKR. New balance: ${balance}")  
    } else if (amount > balance) {  
      println("Insufficient balance.")  
    } else {  
      println("Withdraw amount must be positive.")  
    }  
  }  
}
```

Terminal Output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10\Q3.scala"  
Deposited $200.0. New balance: $1200.0  
Withdrew 150.0 LKR. New balance: $1050.0  
Withdrew 300.0 LKR. New balance: $750.0  
Deposited $300.0. New balance: $800.0  
Transferred 300.0 LKR to target account.  
Account 1 balance: 750.0LKR  
Account 2 balance: 800.0LKR  
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10>
```

Q4

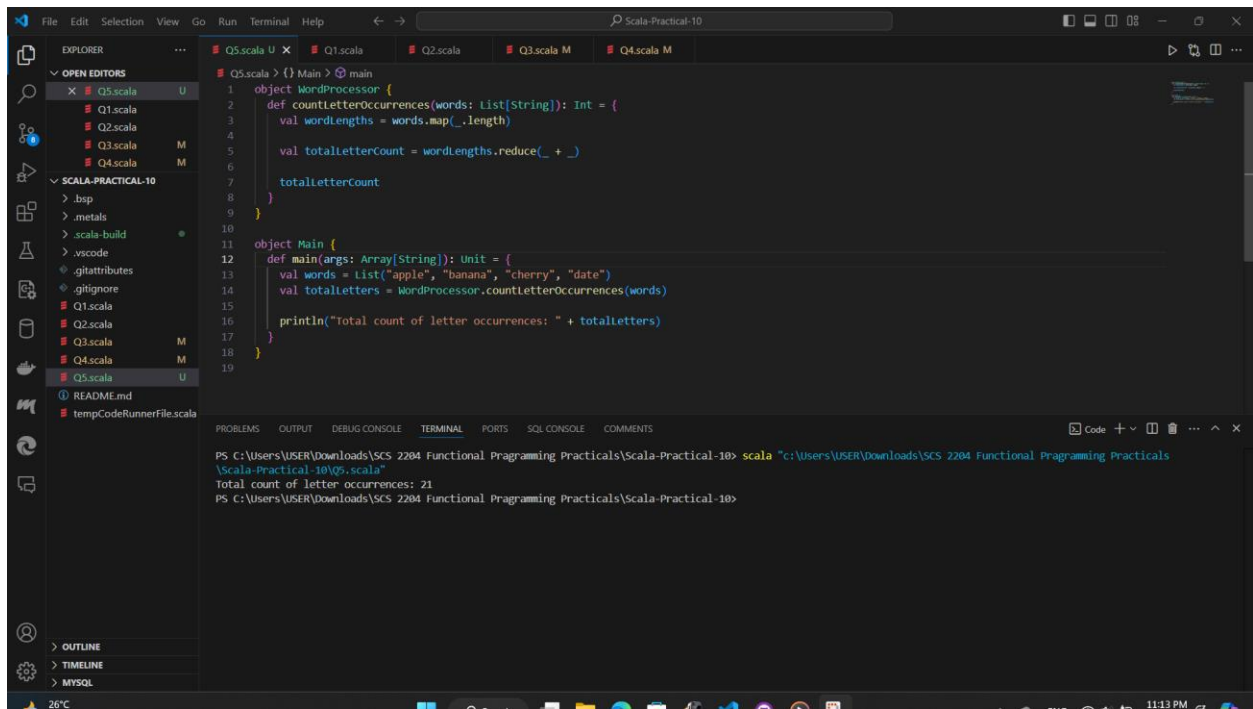
The screenshot shows the VS Code editor with a Scala file named Q4.scala. The code defines an Account class with transfer and applyInterest methods. The terminal output shows the execution of the code, demonstrating the transfer and applyInterest functionality.

```
class Account(private var balance: Double) {  
  def transfer(amount: Double, targetAccount: Account): Unit = {  
    this.withdraw(amount)  
    targetAccount.deposit(amount)  
    println(s"Transferred ${amount} LKR to target account.")  
  } else if (amount > balance) {  
    println("Insufficient balance for transfer.")  
  } else {  
    println("Transfer amount must be positive.")  
  }  
}  
def getBalance: Double = balance  
def applyInterest(): Unit = {  
  if (balance > 0) {  
    balance += balance * 0.05  
  } else {  
    balance += balance * 0.1  
  }  
}
```

Terminal Output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10\Q4.scala"  
Accounts with negative balances:  
Account(balance: -200.0 LKR)  
Account(balance: -50.0 LKR)  
Total balance in the bank: 1250.0 LKR  
Final balances after applying interest:  
Account(balance: -220.0 LKR)  
Account(balance: -55.0 LKR)  
Total balance in the bank after interest: 1300.0 LKR  
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10>
```

Q5



The screenshot shows an IDE window titled "Scala-Practical-10". The Explorer sidebar on the left lists files under "SCALA-PRACTICAL-10", including ".bsp", ".metals", "scala-build", ".vscode", ".gitattributes", ".gitignore", "Q1.scala", "Q2.scala", "Q3.scala", "Q4.scala", "Q5.scala", "README.md", and "tempCodeRunnerFile.scala". The "Q5.scala" file is open in the editor, showing the following Scala code:

```
1 object WordProcessor {
2   def countLetterOccurrences(words: List[String]): Int = {
3     val wordLengths = words.map(_.length)
4
5     val totalLetterCount = wordLengths.reduce(_ + _)
6
7     totalLetterCount
8   }
9 }
10
11 object Main {
12   def main(args: Array[String]): Unit = {
13     val words = List("apple", "banana", "cherry", "date")
14     val totalLetters = WordProcessor.countLetterOccurrences(words)
15
16     println("Total count of letter occurrences: " + totalLetters)
17   }
18 }
19
```

The TERMINAL panel at the bottom shows the command and output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10\Q5.scala"
Total count of letter occurrences: 21
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-10>
```