Vijay's Assignment – Scala4

Task 1

Write a simple program to show inheritance in scala.

Inhertiance in Scala is achieved through "extends" keyword while defining the child class.

- A Parent class is defined below with a simple print statement and a add method.
- Child1 inherits these from Parent. It calls the function add to add 10 & 20.
- Child1 adds a new print method to display the output of this addition
- Child2 inherits all the above by extending Child1. It also overrides the print method from Child1 to display something different.
- Note that "I am inside the parent" is called twice when both ob & ob1 objects are created and they call the print method
- This example shows simple inheritance and multi level inhertiance with method overriding concept

```
package acad_scala2

 class Parent{  /* Parent class with a print statement and a method add to add two numbers */
   println("I am inside the parent ")
   def add(a:Int,b:Int)={
      a + b
   }
}
 class Child1 extends Parent{ /* first level of inheritance where the add method from the parent class is called and result is displayed */
   var c = add(10,20)
   def print()={
   println("I am inside the Child1")
   println("Addition of 10 & 20 from child 1 is: "+c)
   }
 }

 class Child2 extends Child1{ /* Second level of inhertance from child 1 where the print method in Child 1 is overridden to print something different
   override def print()={
     println("I am inside the child2")
     var c = add(40,50)
     println("Addition of 40 & 50 from child2 is:"+c)
     }
 }
 object sct2 {
   def main(args: Array[String]): Unit ={
    val ob = new Child1()
    ob.print()
        val ob1 = new Child2()
    ob1.print()
   }
 }
```

Output

```
<terminated> sct2$ [Scala Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (23-Jan-2019, 12:59:49 PM)
I am inside the parent
I am inside the Child1
Addition of 10 & 20 from child 1 is: 30
I am inside the parent
I am inside the child2
Addition of 40 & 50 from child2 is:90
```

Task 2

Write a simple program to show multiple inheritance in scala

Multiple inheritance is where one Child inherits values from multiple Parents. In Scala multiple inheritance is implemented using traits

- Trait t1 is defined with the method sum1 which prints a line “This will never be printed from sum1”
- Trait t2 is defined with the method sum1 which prints a line “This will never be printed from sum2”
- Trait t3 is defined with the method sum1 which prints a line “This will never be printed from sum3”
- Class fin is defined to extend from the traits t1, t2 & t3
- Class fin overrides sum1 to add two integerts and sum3 to add four integers. Sum2 from t2 is not overridden
- New object ob for class fin calls sum1, sum2 and sum3
- Sum1 and Sum3 displays the correct output where as for sum2 the line “This will never be printed from sum2” is displayed
- This demonstrats multiple inheritance, traits as well as method overriding

```
package acad_scala3
 trait t1{
  def sum1(a:Int, b:Int)={
    println("This will never be printed from sum1")
  }

}
 trait t2{
  def sum2(a:Int, b:Int,c: Int)={
    println("This will never be printed from sum2")
  }
}
 trait t3{
  def sum3(a:Int, b:Int,c: Int,d: Int)={
    println("This will never be printed from sum3")
  }
}
 class fin extends t1 with t2 with t3{
   override def sum1(a:Int, b:Int)={
     println("Sum of the two input number is :"+(a+b)) }
   override def sum3(a:Int, b:Int,c:Int, d:Int)={
     println("Sum of the four input number is :"+(a+b+c+d))}
 }
 object sct3 {
   def main(args: Array[String]): Unit ={
    val ob = new fin()
    ob.sum1(10,20)
    ob.sum2(10,20,30)
    ob.sum3(10,20,30,40)
    }
 }
```

Output

```
<terminated> sct3$ [Scala Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (23-Jan-2019, 1:46:07 PM)
Sum of the two input number is :30
This will never be printed from sum2
Sum of the four input number is :100
```

Task 3

Write a partial function to add three numbers in which one number is constant and two numbers can be passed as inputs and define another method which can take the partial function as input and squares the result.

```
package acad_scala4

object sct4 {
    def main(args: Array[String]): Unit ={
       val addconst: PartialFunction[(Int, Int), Int] = { /* Partial function definition to ADD constant 500 to the two input numbers a & b */
         case (a, b) => a + b + 500
       }
      def functionhigher(f:Int):Unit = {     /* Method that receives an Int as input and calculates the square of the result */
         print(f*f)
       }
      println("Enter the first integer ")
      val a = scala.io.StdIn.readInt()
      println("Enter the second integer ")
      val b = scala.io.StdIn.readInt()

      print("Square of 500 +"+a+"+"+b+"is :")
      functionhigher(addconst(a,b))

    }
}
```

Output

```
<terminated> sct4$ [Scala Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (23-Jan-2019, 7:56:34 PM)
Enter the first integer
10
Enter the second integer
20
Square of 500 +10+20is :280900
```

Task 4

Write a program to print the prices of 4 courses of Acadgild:

Android App Development -14,999 INR

Data Science - 49,999 INR

Big Data Hadoop & Spark Developer – 24,999 INR

Blockchain Certification – 49,999 INR

using match and add a default condition if the user enters any other course.

- Using Scala match function to match the input value to a list of cases
- Displaying “Invalid course number” as the default text.

```
package acad_scala5

object sct5 {
   def main(args: Array[String]): Unit ={
      println("Following are the courses in Acadgild. Provide your choice(1/2/3/4) to know the rates")
      println("1. Android App Development ")
      println("2. Data Science  ")
      println("3. Big Data Hadoop & Spark Developer  ")
      println("4. Blockchain Certification  ")
      def matchacadgild(x:Int):String = x match { /* Use scala match function to evalulate the input value to the corresponding case */
        case 1 => "14,999 INR"
        case 2 => "49,999 INR"
        case 3 => "24,999 INR"
        case 4 => "49,999 INR"
        case _ => "Invalid course number"
        }
      println("Enter your choice as 1/2/3/4")
      val input = scala.io.StdIn.readInt()
      println("The actual rate is: "+matchacadgild(input))
   }
}
```

Output selecting the input value of 3

```
<terminated> sct5$ [Scala Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (23-Jan-2019, 7:30:26 PM)
Following are the courses in Acadgild. Provide your choice(1/2/3/4) to know the rates
1. Android App Development
2. Data Science
3. Big Data Hadoop & Spark Developer
4. Blockchain Certification
Enter your choice as 1/2/3/4
3
The actual rate is: 24,999 INR
```

Output providing an invalid input value of 5

```
<terminated> sct5$ [Scala Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (23-Jan-2019, 7:34:50 PM)
Following are the courses in Acadgild. Provide your choice(1/2/3/4) to know the rates
1. Android App Development
2. Data Science
3. Big Data Hadoop & Spark Developer
4. Blockchain Certification
Enter your choice as 1/2/3/4
5
The actual rate is: Invalid course number
```