Scala SCS 2204 - Functional Programming
Scala Tutorial – 4

1. Develop the function of interest. It consumes a deposit amount and produces the actual amount of interest that the money earns in a year. The bank pays a flat 2% for deposits of up to Rs. 20 000, a flat 4% per year for deposits of up to Rs. 200000, a flat 3.5% per year for deposits of up to Rs. 2000000, and a flat 6.5% for deposits of more than Rs. 200000.

2. Write a Scala program and use a case class to define a shopping cart item. Each shopping cart item should have the following properties, namely, a name, a price, and a quantity bought. Create three shopping cart items for the following items:
   13 vanilla ice cream at Rs3.99 each
   6 chocolate biscuits at Rs 4 each
   7 cupcakes at Rs 7.77 each
   Use an appropriate data structure to store the above-mentioned shopping cart items. Thereafter, define and use a method that will print out all items from a given shopping cart.

   Output:

   ```
   13 vanilla ice cream at Rs3.99 each
   6 chocolate biscuits at Rs 4 each
   7 cupcakes at Rs 7.77 each
   ```

Define also another method that given a shopping cart basket will only output vanilla ice cream products. A generic message, such as, "Found another item", will be the output for all other items.

3 .Write a Scala program to represent a basic car inventory. You should use a case class to model a Car object with a name property of type String, and a price property of type Double. Next, you should use an implicit function that defines a uuid method for each Car object, but without manually modifying the above Car type and definition. In doing so, you can define, say, a CarUUID class which has a constructor parameter for the Car type, and implements a uuid

method. The actual uuid heuristics can be very basic, such as, an output with the combined car name and the car name's hashCode. To verify your uuid method, you can define the following car sample:

- a bmw 3 series priced at 20,000
- a bmw 5 series priced at 50,000
- a vw passat priced at 10,000
- a vw golf priced at 12,000
- a mazda 3 priced at 15,000

Use an appropriate data structure from Scala's collection types to store the above car sample. Finally, you should output each car's uuid method as shown below.

Output:

```
car uuid = bmw 3 series - -2034747624
car uuid = bmw 5 series - 1450873046
car uuid = vw passat - 44703299
car uuid = vw golf - 790852193
car uuid = mazda 3 - 846423990
```

4. Write a program in PatternMatching, takes the integer input from the command line. Based on the input, write a code using match to print Negative/Zero is input when input is less than or equal to 0.Print Even number is given when input is even, and print Odd number is given when input is odd.

5. Write a Scala program which defines a method named "toUpper" and it accepts a String as input parameter that is then formatted to upper case as output. Define another method named "toLower" which accepts a String as input parameter and formats the input to lower case as output. Define another method named "formatNames" which also has an input String called "name". This method however has a parameter group which accepts a function with an input of type String and also outputs a String. This particular function will be used to apply the given format to the "name" input. You can use the test inputs for, say, "Benny", "Niroshan", "Saman" “Kumara”, and make sure that the output is as shown below.

Output:

```
BENNY
NIroshan
saman
KumarA
```