#1 : Task Requirement

WAP to enroll admission for College Students by defining following functions.

1. enrollStudent()
   read student name, highest qualification, scored marks, university name
   create object array of these details using “Object” class from lang package.
2. validateInput()
   function takes argument of type “Object” array and check each value is entered correctly or not.

   Validation Requirement

   Name should not contain symbols and numbers, qualification should be either [ssc,hsc,graduate,post graduate], scored marks should be in 1 to 100, university name should not contains numbers and symbols except .(dot).

   Function should return following error code by occurrence for wrong input

Error Code	For
-1	Everything correct
101	Invalid name
102	Invalid Qualification
103	Invalid Scored Marks
104	Invalid University Name

   In function enrollStudent() call function validateInput() and prompt error message.

Sample Input1:

Enter Student Name :**Rishi**
Enter Highest Qualification :**Hsc**
Enter Scored Marks :**87.45**
Enter University Name :**Hyderabad**

Sample Output1:

Student Registration Successful

Sample Input2:

Enter Student Name :**vidhi**
Enter Highest Qualification :**graduate**
Enter Scored Marks :**102.32**
Enter University Name :**Chennai003**

Sample Output2:

Invalid Scored Marks

#2 : Task Requirement

WAP to create LinkedList as “myvehicles” which adds objects of class “Vehicle”
Design class Vehicle as follow,
Private variables : vehid, veh_name,fuel_cap,top_speed
functions :
a. constructor : to read parameters vehno, name, fuel_cap and top speed
b. toString() : override method to return vehical information

A. Create another list from “myvehicles” which only adds that vehicles which has fuel_cap greater than 25 Lit.
B. Sort Vehicles by their Top Speed in Ascending Order. (Use Comparable Implementation)

Sample Input1:

new Vehicle(394,"Bullet",12.00f,160.00f)
new Vehicle(395,"Kawasaki",15.00f,190.00f)
new Vehicle(396,"Honda",8.00f,100.00f)

new Vehicle(397,"Moped",3.00f,70.00f)
new Vehicle(398,"Car",30.00f,200.00f)

note: add new instance of class objects to LinkedList

Sample Output1:
myvehicles
394 Bullet 12.0 160.0
395 Kawasaki 15.0 190.0
396 Honda 8.0 100.0
397 Moped 3.0 70.0
398 Car 30.0 200.0

anotherlist
398 Car 30.0 200.0

sorted myvehicles
397 Moped 3.0 70.0
396 Honda 8.0 100.0
394 Bullet 12.0 160.0
395 Kawasaki 15.0 190.0
398 Car 30.0 200.0

#3 : Task Requirement
WAP to define a single array where you store 10 values of type string , integer and float.
Define a following function having argument of type array
a. getIntegerValues():
function determines which values are integers and returns new array contains integers
b. getFloatValues():
function determines which values are float and returns new array contains float
c. getIntegerString():
function determines which values are String and returns new array contains String

Sample Input1:
{213,"Welcome",true,"Hello",11,8.2f,0.35f,false,"India",190}

Note: create this array in main() method

Sample Output1:
All int from array
213 11 190
All float from array
8.2 0.35
All String from array
Welcome Hello India

Sample Input2:
{null,0,-1233,"wrong","java",10.3344,true,"false",100-45,43.00f-3.4f}

Note: create this array in main() method

Sample Output2:
All int from array
0 -1233 55
All float from array
39.6
All String from array
wrong java false

#4 : Task Requirement

WAP to create following classes with **default** and **parameterized** constructor

1. class "Labor"
   variables : labor name, age, town, contactno *[variables should be private]*
2. class "Lorry"
   variables : lorry_name, vehicle_number, weight_of_goods_in_tons *[variables should be private]*
3. class "Contracts"
   variables : contract_id,contract_person_name,labor_obj, lorry_obj *[variables should be protected]*

Define **toString() method** in class "Labor" , "Lorry" and "Contracts" which returns formatted output of String.format() method

Define main() function as follow,

1. create 5 "Labor" class object and must call parameterized constructor for 4 times and default constructor for 1 time
2. create 3 "Lorry" class object and must call only parameterized constructor
3. create 6 objects of class "Contracts" and call parameterized constructor at least for 4 times

Sample Output:

Labor

John	37	California	+1 9090090909
Smith	41	Moscow	+7 9802389823
Peter	29	Bijing	+86 238982375
Michael	28	Fransisco	+1 2893248973
#Unkwon_name	0	#UnkownTown	#UnknownContactNo

Lorry

Panel truck	83774	20
Flatbed truck	9929	65
Dump truck	10084	105

Contracts

91	James	Smith	Panel truck	83774
92	Robert	Peter	Panel truck	83774
93	Richard	Smith	Dump truck	10084
94	Paul	Michael	Flatbed truck	9929
0	0	# UnknownLabor	# UnknownLorryName	0
0	0	# UnknownLabor	# UnknownLorryName	0