PRACTICAL IN CLASS

Question 1:

Do not pay attention to real meaning of objects, variables and their values in the questions below.

Write a class named **Engine** with the following information:

Engine
-designer:String -power:int
+Engine() +Engine(designer:String, power:int) +getDesigner():String +getPower():int +setPower(power:int):void

Where:

- Engine() - default constructor.
- Engine(designer:String, power:int) - constructor, which sets values to designer and power.
- getDesigner():String – returns a string s, which is obtained by taking the 3 first characters of string and lowercase the first character in the s string.
- getPower():int – return power.
- setPower(power:int):void – update power.

Do not format the result.

The program output might look something like:

Enter designer: Tom123 Enter power: 10 1. Test getDesigner() 2. Test setPower() Enter TC (1 or 2): 1 OUTPUT: tom	Enter designer: Tom123 Enter power: 10 1. Test getDesigner() 2. Test setPower() Enter TC (1 or 2): 2 Enter new power: 12 OUTPUT: 12

Question 2:

Write a class **Robot** and a class **SpecRobot** extending from **Robot** (i.e. Robot is a superclass and SpecRobot is a subclass) with the following information:

Robot
-label:String -type:int
+Robot() +Robot(label:String, type:int) +getLabel():String +getType():int +setLabel(label:String):void +toString():String

Where:

- getLabel():String – return label.
- getType():int – return type.
- setLabel(label:String):void – update label.
- toString():String – return the string of format: **label, type**

SpecRobot
-step:int
+SpecRobot() +SpecRobot(label:String, type:int, step:int) +toString():String +setData():void +getValue():int

Where:

- toString():String – return the string of format:
  label, type, step
- setData():void – Insert step into the 2nd character of the label.
- getValue():int – Check if the type < 3 and the label contains 'A' character then return step, otherwise return step + 2.

The program output might look something like:

Enter label: asimo Enter type: 1 Enter step: 2 1. Test toString() 2. Test setData() 3. Test getValue() Enter TC (1,2,3): 1 OUTPUT: asimo, 1 asimo, 1, 2	Enter label: asimo Enter type: 1 Enter step: 2 1. Test toString() 2. Test setData() 3. Test getValue() Enter TC (1,2,3): 2 OUTPUT: a2simo, 1	Enter label: Asimo Enter type: 1 Enter step: 3 1. Test toString() 2. Test setData() 3. Test getValue() Enter TC (1,2,3): 3 OUTPUT: 3	Enter label: Asimo Enter type: 3 Enter step: 3 1. Test toString() 2. Test setData() 3. Test getValue() Enter TC (1,2,3): 3 OUTPUT: 5