# Task 1

1. **Positive Integer Code**

**import** java.util.Scanner;

**public** **class** Task1A {

**public** **static** **void** main(String[] args) {

**double** even = 0, odd = 0, sum = 0;

**int** counter = 0, input = 0, large = Integer.***MIN\_VALUE***,

small = Integer.***MAX\_VALUE***;

System.***out***.print("Enter series of positive Integers (enter 0 to exit the program): ");

Scanner in = **new** Scanner(System.***in***);

**while** ((input = in.nextInt()) != 0) {

small = Integer.*min*(small, input);

large = Integer.*max*(large, input);

sum += input;

counter++;

**if** (input % 2 == 0) {

even++;

} **else** {

odd++;

}

}

**if** (counter > 0) {

**double** average = sum / counter;

System.***out***.println("Smallest integer is: " + small);

System.***out***.println("Largest integer is: " + large);

System.***out***.println("Number of entered integers is " + counter);

System.***out***.println("Average value is: " + average);

} **else** {

System.***out***.println("No entered data.");

}}}

1. **Guessing Code**

**import** java.util.Scanner;

**public** **class** Task1B {

**public** **static** **void** main(String[] args) {

**int** secretNumber;

secretNumber = (**int**) (Math.*random*() \* 999 + 1);

Scanner keyboard = **new** Scanner(System.***in***);

**int** guess;

**do** {

System.***out***.print("Enter a guess (0-1000): ");

guess = keyboard.nextInt();

**if** (guess == secretNumber)

System.***out***.println(" C!");

**else** **if** (guess < secretNumber)

System.***out***.println(" L!");

**else** **if** (guess > secretNumber)

System.***out***.println(" H!");

} **while** (guess != secretNumber);

}

}

# Task 2

1. Objects, Class and Methods

* Objects is an instance of class having behaviors and states. For instance: A dog having states of name, color, breed and behaviors of barking, shaking the tail, eating. An object is an instance of a class (W3schools.com. 2021. , [Accessed 30 March 2021].). Making the comparison amid the real world objects such as dog, cat, cow and cars they have similar features as the software program having states and behavior where behavior is stored in the fields where behavior is shown through methods (Wertz, H, 2015.).
* Methods are collections of declarations which are assembled organized in performing a process (Wertz, H, 2015.). The method only executes when it is called. Method ought to be declared inside a class and used in reusing the codes (Sarcar, V. and Mullick, A., n.d. , n.d.). Method comprises of method body and method header as shown in the syntax below:

modifier returnType methodName (List of Parameter) {

// method body

}

Where:

Modifier defines access type of methods and method body states what methods do with declarations.

* Class is a blueprint which defines the state which the object supports or where discrete objects are formed (Wertz, H, 2015.). It is as illustrated below:

public class Puppy {

String color;

int age;

String breed;

void growling () {

}

void starving () {

}

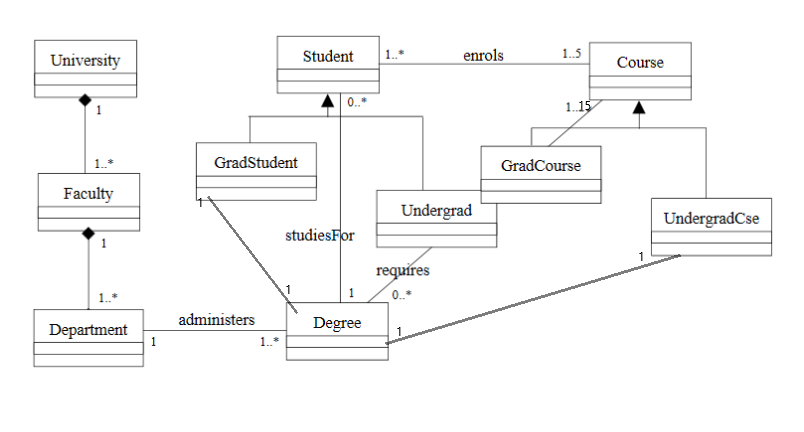
void snoozing () {

}

}

In the above, growling, starving and snoozing are methods

The class may comprise of any of the local, instance and class variables. Local variables are defined in the method, blocks or constructors (Tokoro, M., n.d. , n.d.). Also declared as well as initialized inside methods which later destroyed upon the completion of the methods. Instance variables are inside a class on the other hand outside any of methods (Wertz, H, 2015.). They are initialized upon the instantiation of the class, also accessed from within any constructors, methods of the specific class. Class variables are stated inside a class and outside any methods, using static keywords (W3schools.com. 2021. , [Accessed 30 March 2021].).



# Task 3

**class** Person {

**protected** String name; // name of the person

**protected** **int** age; // person's age

**protected** String gender; // "M" for male, "F" for female

**public** **static** **void** main(String[] args){

Person myInfo = **new** Person("John Wilson ",97,"male");

**var** name = myInfo.getName();

System.***out***.println("Name is: "+myInfo.getName());

System.***out***.println("Age is: "+myInfo.getAge());

System.***out***.println("Gender is: "+myInfo.getGender());

}

/\*\* GETTERS \*\*/

**public** String getName(){

**return** name;

}

**public** String getGender(){

**return** gender;

}

**public** **int** getAge(){

**return** age;

}

/\*\* SETTERS \*\*/

**public** **void** setName(String name){

**this**.name = name;

}

**public** **void** setGender(String gender){

**this**.gender = gender;

}

**public** **void** setAge(**int** age){

**this**.age = age;

}

**public** Person(String n, **int** a, String g) // constructor

{

name = n; age = a; gender = g;

}

**public** String toString() {

**return** name + ", age: " + age + ", gender: " +gender;

}

}

**class** Student **extends** Person {

**protected** String idNum; // Student Id Number

**protected** **double** averageMark; // grade point average

**public** Student(String n, **int** a, String g, String id, **double** average) {

**super**(n, a, g); // inherit from person

//initialize the new instance variables for students

idNum = id;

averageMark = average;

name = n;

age = a;

gender = g;

}

}

**class** Teacher **extends** Person {

**private** String subject;

**private** **double** salary;

**private** String myName;

**private** **int** myAge;

**private** String myGender;

**public** Teacher(String n, **int** a, String g, String s, **double** sal ) {

**super**(n, a, g);

myName=n; myAge=a; myGender=g; subject=s; salary=sal;

}

/\*\* GETTERS \*\*/

**public** String getMyName(){

**return** myName;

}

**public** String getSubject(){

**return** subject;

}

**public** **double** getSalary(){

**return** salary;

}

**public** String getMyGender(){

**return** myGender;

}

**public** **int** getMyAge(){

**return** myAge;

}

/\*\* SETTERS \*\*/

**public** **void** setMyName(String myName){

**this**.myName = myName;

}

**public** **void** set(**double** salary){

**this**.salary = salary;

}

**public** **void** setGender(String myGender){

**this**.myGender = myGender;

}

**public** **void** setMyAge(**int** myAge){

**this**.myAge = myAge;

}

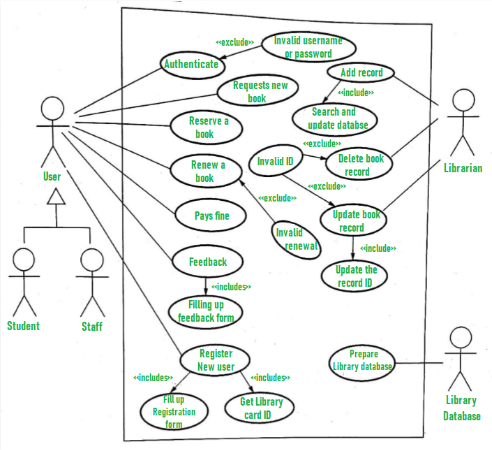
**public** String toString() {

**return** myName + ", myAge: " + myAge + ", myGender: " + myGender + ", subject: " + subject + ", salary: " +salary;

}

}

# Task 4



# References

Sarcar, V. and Mullick, A., n.d. , n.d. In: *Interactive object-oriented programming in Java..* s.l.:s.n.

Tokoro, M., n.d. , n.d. In: *Object oriented programming..* s.l.:s.n.

W3schools.com. 2021. , [Accessed 30 March 2021].. Object oriented programming.. In: *Java OOP (Object-Oriented Programming). .* s.l.:[online] Available at: <https://www.w3schools.com/java/java\_oop.asp>.

Wertz, H, 2015.. Object -oriented Programming with Smalltalk.. In: s.l.:Elsevier Science..