Question 1

a)

The principle of data hiding helps the programmer to build secure programs.

Through inheritance, we can eliminate code and extend the use of existing classes.

Reusability.

Easy to understand.

Software complexity can be easily managed.

OO Systems can be easily upgraded from small to large system.

b)

A class is a user defined prototype for an object that defines a set of attributes that characterize any object of the class. A class is a blueprint of an object.

An object is a unique instance of data structure that's defined by its class. An object comprises both data members and methods.

Ex : If 'Car' took as a class, BMW, Toyota, Honda are objects of 'Car' class.

c)

1.First identify the data in the objects as well as the process or actions that can be performed on that data.

2.Encapsulate their data and the processes that act on those data.

3.Using information hiding method hide private data of an object from other objects.

4.Determine the relationship between objects.

5.Design the algorithms for methods, using structured design.

6.Delevop the program from the algorithm.

d)

Class

Object

Abstraction

Encapsulation

Polymorphism

e)

Super class – is a base class of an inheritance relationship. In inheritance relationship new classes take on the properties of existing super class.

Ex : In Parent and Child inheritance relationship, Parent is the super class (base class) and Child is the subclass.

Constructor - is a special method of a class in OOP that initializes the object when it is declared. A type of member function that is automatically executed when an object of the class is created is known as Constructor.

Ex : In Employee class, Employee() is the default constructor.

Function Overloading – is having multiple functions with same name, but with different types of parameters. In overloading the parameters of each function should be different. The return type and the function name have to remain the same.

Ex : void print();

void print(char msg[]);

void print(char msg[], int no);

f)

Classes	Objects	Attributes
Customer	Smith	Name, Address, Contact Number
Item	Books, Magazines, CDs, Newspapers	Price, Description
Purchase	“C++ : How to Program”, “Business Today”	Date, items, amount

Question 2

a)

Inheritance < ----------

Composition **<>**----------

Aggregation <>----------

Association ----------

Dependency <- - - - - - - -

b)

i) Inheritance – is a relationship between two or more classes where derived class (sub class) inherits properties of pre-existing (base) classes.

ii)

```
class Picture{
        protected :
                string title;
                double price;
        public :
                void updatePrice();
};

class Photograph : public Picture{
        private :
                string photographer;
                string camera;
                int speed;
                int aperture;
        public :
                void alterConstrast();
};
```

```
class Painting : public Picture{
        private :
                string artist;
                string type;
                int owner;
        public :
                void printProvenance();
};
```

c)

i)

```
Box :: Box() {
        length = 0;
        width = 0;
        height = 0;
}
```

ii)

```
Box :: Box( int l, int w, int h) {
        length = l;
        width = w;
        height = h;
}
```

iii)

```
int main() {
        Box box1, box2( 5, 2, 3);
        Box *box3;
        box3 = new Box;
        cout << “Volume : “ << box2.findVolume() << endl;
        box3 -> setLength(10);
        box3 -> setWidth(7);
        box3 -> setHeigth(3);
        return 0;  }
```

Question 3

a)

Redundant

Meta Language

An event or an operation

Outside scope of system

An attribute

b)

i)

bike - class

details – meta-language

bike number – an attribute

type - an attribute

size - an attribute

make - an attribute

model - an attribute

daily charge rate - an attribute

deposit - an attribute

customer - class

rent - an attribute

number of days - an attribute

rent transaction - class

start date - an attribute

estimated duration - an attribute

unified receipt - outside the scope of system

extra - an attribute

total amount - an attribute

receipt – outside the scope of system

state of bike - an attribute

classes – customer, bike, rent transaction

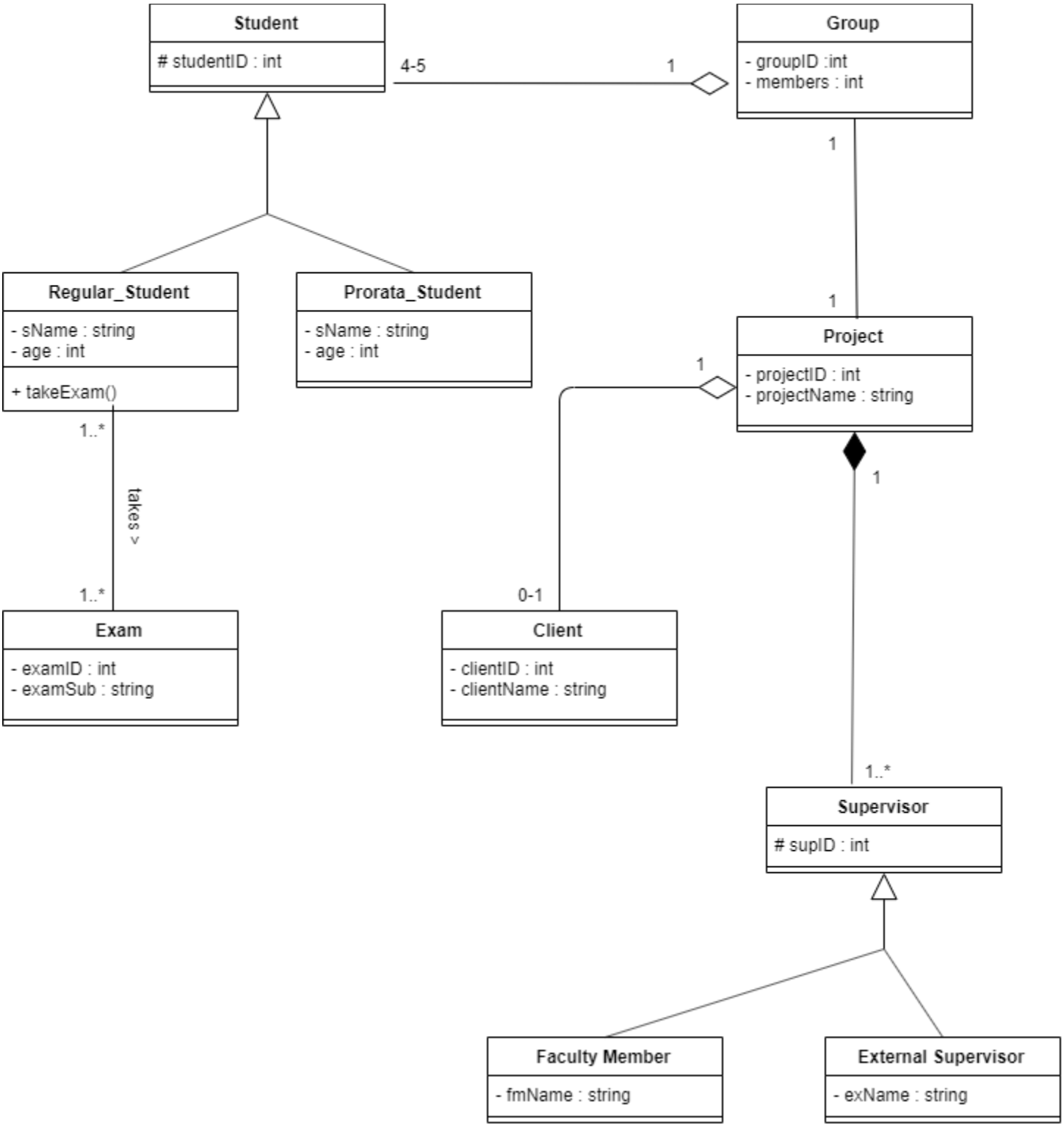
ii)

Customer	
Responsibilities	Collaboration
Store details of customers	
Store details of past transaction	Rent transaction
Rent bike	Bike
Return bike	Bike
Record payment	
Print receipt	

Bike	
Responsibilities	Collaboration
Store details of bike	
Store details of state of bike	

Rent Transaction	
Responsibilities	Collaboration
Record the details	Bike, customer

Question 4



h.g.d.sandakalum@gmail.com