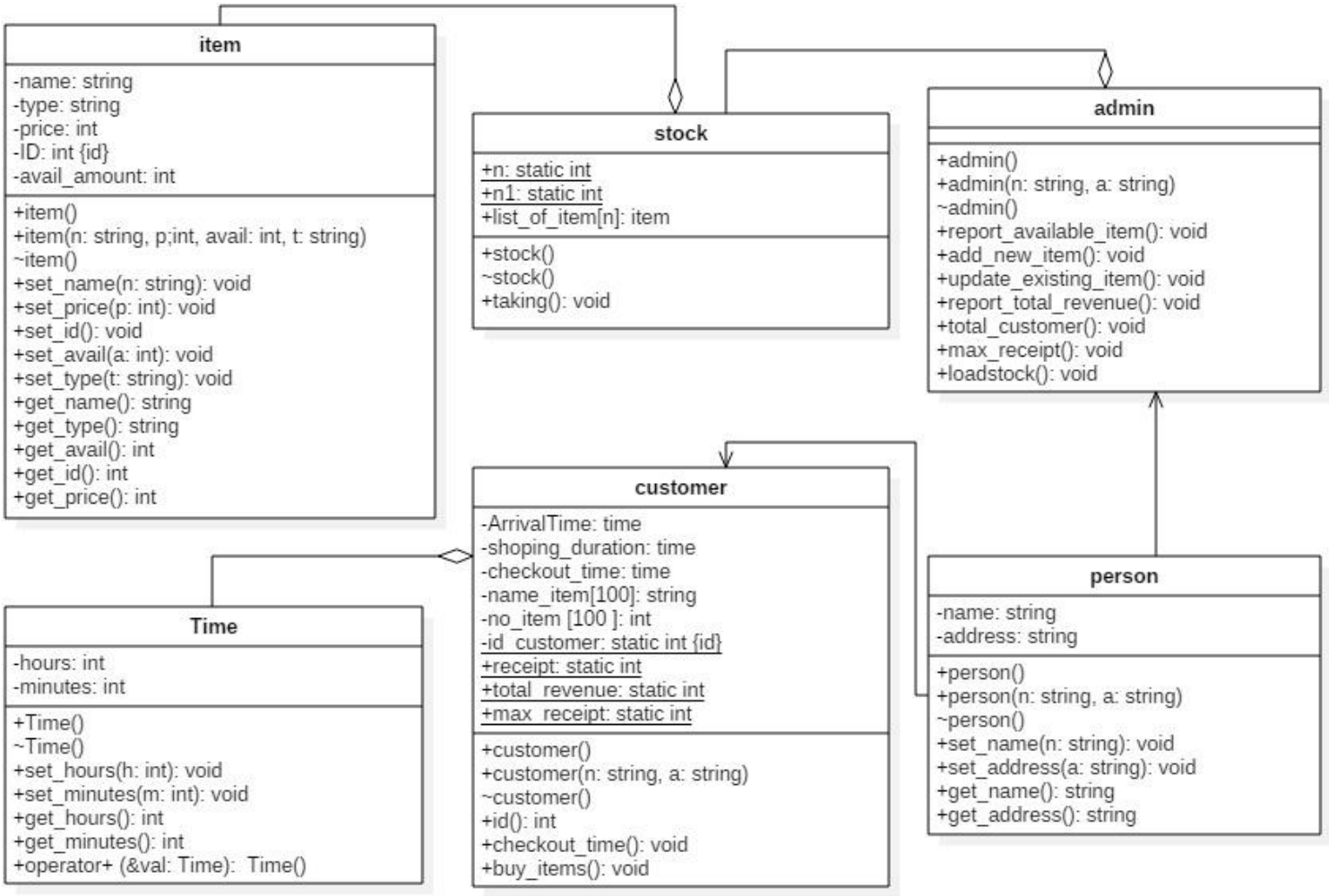
Faculty of Computers and information
Cairo University

OOP Project

Names	IDs	Groups
1- Ahmed Saleh Mahmoud	20160016	7
2- Mohammad Ahmad Mostafa	20160192	7
3- Mohamed Ramadan Hussein	20160205	8
4- Mohamed Atef Mohamed	20160211	8
5- Mohamed Abdelbaset	20160212	8

Faculty of Computers and information
Cairo University

Project UML



Faculty of Computers and information
Cairo University

Class Stock:-

Class Stock	To store and represent list of items
Static int n	Size Of array of the items
Static int n1	Size of exist item of array
Void taking ()	To take items from the file
Item *List_of_item	Store list of item
Stock()	Default Constructor
~Stock()	Default Destructor

Class Time:-

Class time	To calculate time and represent it
Int hours	Count of hours
Int minutes	Count of minutes
time()	Default Constructor
~time()	Default Destructor
Set_hours (int h)	To calculate arrival time per hours
set_minutes (int m)	To calculate arrival time per minutes
get_hours (int)	To print shopping duration per hours
get_minutes (int)	To print shopping duration per minutes
Operation+ (Time &val)	To sum to arrival time and shopping duration for user

Class item:-

Class item	It's used to set price , type , available amount and id for each item
String name	Variable of name
Int price	Variable of price of item.
Int ID	Variable of id of item.
Int avail_amount	Variable of available amount
String type	Variable of type of item
Item(string n, int p, int avail, string t)	Constructor for class item take parameters like string , int , int and string to set them .
Item()	Constructor for the class.
~Item()	Destructor for the class.
Void set_name (string n)	Function to set name
Void set_price(int p)	Function to set price
Void set_id()	Function to set id
Void set_avail(int a)	Function to set available amount
Void set_type(string t)	Function to set type
String get_name()	Function to get name
String get_type()	Function to get type
Int get_avail()	Function to get available amount
Int get_id()	Function to get id
Int get_price()	Function to get price

Faculty of Computers and information
Cairo University

Class Person:-

class person	**A class stores and print name and address of customer or admin**
string name	**A variable stores name of admin or customer**
string address	**A variable stores address of customer or admin**
person()	**An empty constructor**
person(string n, string a)	**A constructor stores name and address of customer or admin**
~ person()	**A destructor**
set_name(string n)	**A function stores name of customer or admin**
set_address(string a)	**A function stores address of customer or admin**
get_name()	**A function print name of customer or admin**
get_address ()	**A function print address of admin or customer**

Faculty of Computers and information
Cairo University

Class Admin "inherits from person":-

class admin	**Prints list of products ,add new item ,update item ,total revenue ,number of customers ,maximum revenue**
admin()	**Is an empty constructor**
admin(string n,string a)	**Is a constructor sends parameters to functions of person class**
~ admin	**A destructor**
report_availabe_item()	**Calls function in stock class to print list of available products**
add_new_item()	**Calls function in stock class to add new item in list of products**
update_existing_item()	**Calls function in stock class to update item in list of products**
report_total_revenue()	**A function print total revenue**
total_customer()	**Prints numbers of customers**
max_receipt()	**Prints maximum revenue of customer**
Void loadstock ()	**To read items from the file**

Faculty of Computers and information
Cairo University

Class Customer "inherits from person":-

Class customer	**Class have an array of items and allow the customer to buy items, calculating (arrival, shopping_duration and check_out time), have a stock object, calculating the max receipt of all customers, put a unique ID for each customer and calculate the total revenue of all customer**
Time ArrivalTime ,shoping_duration ,checkout_times	**Objects of time for (arrival time , shopping duration time and checking_out time for the customer**
int no_item[100]	**Array to set the sold items for each customer**
static Int receipt	**To calc. the sum of prices for each customer**
item items[100]	**Array of objects with 100 grocery item**
static int id_customer	**To change every customer ID**
static int total_revenue	**Initial value of 0 to calculate the sum of all revenues**
static int max_receipt	**Initial value of 0 to compare the revenue of each customer**
customer();	**Default constructor**
customer(string,string)	**Parametrized constructor because customer inherits from Person**
int id()	**Function to return the ID of the customer from Person**
void checkout_time()	**Function to calculate the check_out = (arrival + duration)**
void buy_items()	**Function to select items from the stock**
~customer()	**Default destructor**