Independent University, Bangladesh
Course ID : CSE213
Course Title : Object Oriented Programming-1
Midterm examination ; Date : 22-08-2020
Section : 03
ID : 1611042 Name : Fahim Abdullah Sajid

Ans 1) (a) Some of the logically acceptable goals for the user type "Manager" :

~~Goal 1 :- Check whether the necessary products are required or not.~~
Goal 1 :- Check whether all the ~~er~~ staff are present or not.
Goal 2 :- Check stock for neccessary products or foods.
Goal 3 :- Create monthly report.
Goal 4 :- Add ~~Employee or~~ staff account profile.
Goal 5 :- Supervise kitchen process.

(b) Events for Goal 4 :- Add staff account profile.

e1 :- Login into the restaurant software
~~e2 :- An A page~~
e2 :- Go to "add staff" option and click
e3 :- A page apear with the required fields to be filled up about the staff.
Required fields are : Name, Address, Phone no. and e-mail.
e4 :- Press "save" option to save information.
e5 :- The information is recorded in the database.

(c) ~~Pub~~

```
public class Person {
    protected String name, Address, Phone No. and e-mail;

public Person(){
        this.name = Address = PhoneNo == e-mail;
public Person (String name, string Address, Int PhoneNo, string e-mail);
        this.name = name;
        this.Address = Address;
        this.PhoneNo = PhoneNo;
        this.email = e-mail;

public String getname(){
        return name;
    }
public void setname(string name) {
        this.name = name; }
public string getaddress(){
        return address;
    }
public void ~~get set addr~~ setaddress(string address) {
                this.address = address; }
public ~~voi~~ string getPhoneNo() {
                return PhoneNo; }
public void setPhoneNo(~~string~~ Int PhoneNo) {
                this.PhoneNo = PhoneNo; }
public string getemail(){
                return e-mal; }
public void set e-mail(Int e-mail) {
                this.e-mail = e-mail; }
public void ~~setper~~ setPersonInfo(){
                scanner S = new Scanner(system.in);
```

next page →

```
import java.util.scanner;
public class Staff extends Person{
    private string name.
        private int ~~enteringtime~~ time ; private string present;
        public void setStaffInfo ~~()~~ (){
            Scanner s = new Scanner (System.in);
            Sout ("Staff infor will be input by the manager")
            time = s.nextInt();
            SetStaffInfo ();
```

Ans 2)

```
Class Array{
protected int[] intData;
public void setArray(int[] a){
        intData = a;
    }
public void showArray() {
        System.out.print("Array will be shown");
        for (int i=0; i<intData.length; i++){
                cout (intData[i]);
            if (i < intData.length-1)
                System.out.print(" ");
            cout ("slam")
```

```
Class MyArray extends Array {
private int sum;
 sum = s.sum;
 public void setMyArray()
 Random r= new Random();
    int[ ] arr = new int[r]
 for (int i=0; i<arr.length; i++){
            Arr[i] = r.nextInt();
            MyArray.setArray();}
```

```
public void showMyArray(){
        MyArray.setArray();
        Sout("  ")

~~public void~~
 public Int getArray(){
        return MyArray;
public void setArray(Int[]) {
        This.setArray = setArray; }

Sout("how many rows")
rows = sc.nextInt();
twoD = new int[rows][];
for (i=0; i<twoD.length; i++){
        twoD[i] = new int[2];
    for (j=0; j<twoD[i].length; j++)
```

~~Ans 3)~~

~~Class Engine {~~

~~private~~

~~public:~~

private:

float horsepower; string engType; int noOfCylinder;

public:

~~horsepower() ;~~

horsepower(); engType; noOfCylinders

friend ~~float~~ horsepower; string engType; int noOfCylinders;

friend int (Engine, Truck);

int add;

Ans 3) Class Engine {

Public:

float horsepower; string engType, int noOfCylinder;

friend int ~~Engine(Engine~~ Eng(Engine);

Class Truck {

Public: Engine eng; string model, truckType; float price;

friend int Trk(Truck);