Date :

ID: 1911116
Name : Fatin Ishraq Ahammed

CSE213 Midterm
Sec - 3 Part - 1

Q1/a) Project Name: Simulating operations of a Catering Service serving different airlines at an airport.

Users	Goals
1. CEO	• Check feedback of the work done
2. Chef	• Check ~~Status of~~ the orders
~~3. Procurement Manager~~	•
3. Supplier	• Get list of ingredients to supply
4. Transport Manager	• Receives delivery request

b) User Chef:
Goal: Check the orders
Workflow:
1) Enter User ID and Password
2) Dashboard for chef appears
3) Navigate to the Food Menu option
4) Validate the number of Items to be cooked

c)

~~Class Check & User Type {~~

```
Class    login {
private :
    ~~int id;~~ string password, id ;

public :

string ~~fun~~ checkId ( ) {

        cout << "Enter ID: " ~~n~~ ";
        cin >> id ;  cin. ignore ( );
        ~~getline (cin, password);~~

        cout << "Enter Password: " ;
        get line (cin, password) ;

        return id ; }
};

Class   User Type {

private :
        String idType; ~~pass~~
        login check;
public:
    void verify User ( .

        id Type = check. checkId ( ) {
```

if (idType == AssignedChefID)

{ Call the class for chef's dashboard };

Class Chef {

private:

string foodItem, ingredients;

int numOrders; ~~ingredients~~

public:

void setItem () {

Take input; Count the number of food items on the menu and assign the total number to 'numOrders'

Take input; Count number of ingredients there are

}

int ~~showOrder~~ showTotalOrders () {

return numOrders; }

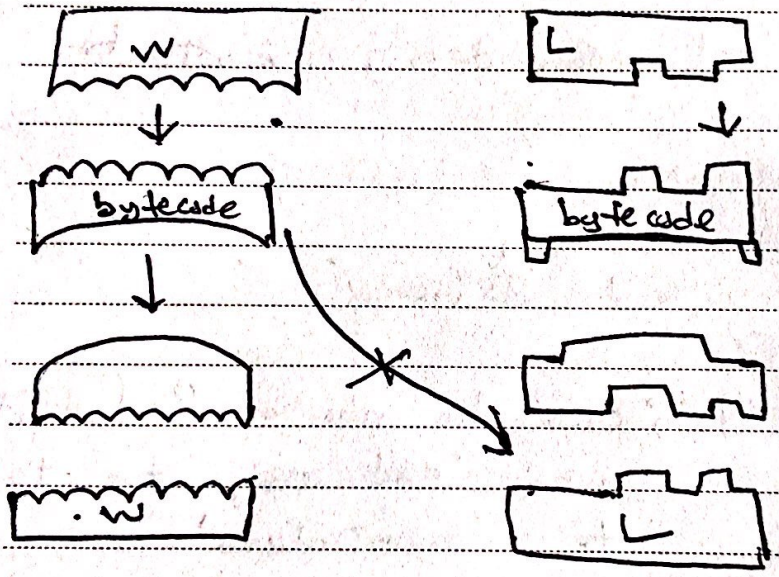
string showIngredients () {

return ingredients; }

Q2/ a) Java ~~has~~ is platform independent because it has its own interpreter and representation of compiled machine code.

The extension is .class.

byte code generated in window will only work in windows. The same applies for other OS.



As explained by the diagram, the shape of the boxes for the byte codes are independent of for the platforms.

b)

b) If a class is public but the method is package, it means that method cannot be used outside of the package.

Example: ~~public class Main {~~

```
package mainpkg;
import . Calculator;
~~public~~
public class Main {

    Calculator use;

    void numbers () {
        use . add ()
        use . isEquation () }
```

will show error →

```
package Calculator;

public class Calculator {

    package void isEquation () { };

    public void add () { };
```

c) Yes they are the same in terms of definition.

asif is the object of the class Student both in Java and C++.

Example

In C++

```
class Student{    // Class is initiated
    private and public variables and methods
    initiated
};
```

In Java

```
public class Student{    // public class initiated
    methods and variables initialize
}
```

To call the methods in main, we have to create the class object in main and in this case:

C++

```
int main () { Student asif ; }
```

Java

```
public class main { student asif ; }
```

Part - 2

Date :

1.

```
class Array {

    int *dataPtr;
    int noOfElements; float avg;

public:

    void setArray(int s, int max) {

        int a[s]; int sizeof = s; float avg=0;

        for (int i=0; i<sizeof; i++)
        { if (rand %100 < max)
            {
                a[i] = rand %100;
            }
        }
    };

Class Array of Arrays { {

    Array* ptrToArrays;
    int noOf Arrays; float avg;

public:
void setArrays of Arrays (int n, int max) {

        ptrToArrays.SetArray(n, max);

    for (int i=0; i<n; i++)
    { int temp += a[i]; }
```

~~void showArrayOfArrays~~

~~float~~ int & avg = temp / &n;
return *this;
}

void showArrayofArrays () {

cout << "Size of obj. no of Arrays is randomly set to: " << n << endl;

cout << "Upper limit of random values is: " << max;

for (int i=0; i<n; i++) {

cout << "Size of the Array of obj. ptToArrays[" << i << "] is randomly set to: " << a[i] << endl;

}
}
};