Question1: Consider the class Item and its subclasses (in page 8) to answer the following Multiple Choices (7 Pts.)

-2.5

1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	D	B	C	B	A	A	C	C	D	A	A	B	D

1. What are the lines where `this` can be deleted without affect code execution?

   A. Class `Medicine` line 12 and 13
   B. Class `Medicine` line 7 and 17
   C. Class `Medicine` line 28 and class `Vitamins` line 7
   D. Class `Medicine` line 7 and class `Vitamins` line 12

2. What are the lines where `super` can be delete without affect code execution?

   A. Class `Medicine` line 11 and class `Vitamins` line 6
   B. Class `Medicine` line 17 and class `Vitamins` line 11
   C. Class `Medicine` line 28 and class `Vitamins` line 11
   D. Class `Vitamins` line 17 and 18

3. If the implementation of `Vitamins` constructer changed to

```
public Vitamins(String name, int p,
String usage, int w)
{   this.year=w;    }
```

super();

The compiler will produce an error?

   A. True
   B. False

default

it will call Medicine constructor

4. If I removed method `computeDose(int age)` from `Medicine` and add it to `Vitamins` the compiler will produce an error because:

   A. `getName()` method is not in `Medicine`.
   B. `Vitamins` class is a direct subclass of `Medicine`.
   C. `Medicine` is not abstract.
   D. `Vitamins` class is not a direct subclass of `Item`.

5. If the implementation of `print()` method in class `Vitamins` changed to

```
public void print()
{
System.out.print(super.name+"
"+super.price+" "+super.usage);
System.out.println(" "+this.year);
}
```

private

The compiler will produce an error?

   A. True
   B. False

6. Can I change the header of `computeDose()` method in class `Medicine` to

```
public final int computeDose(int
age)
```

   A. Yes
   B. No

because we used it once and didnt overrid it

Scanned with CamScanner

name: Centrum Price: 10 usage: poor diet

abstract

7. What is the output of following Java program?

```
public class Test
{public static void main (String[] args)
 {
Item arr[] = new Item [2];
arr [0] =new Item ("Augmentin");
arr [1] =new Medicine ("Panadol",10,"
headache");
for(int i = 0; i<2 ; i++)
System.out.print(arr[i].getName()+" ");
 }}
```

A. Augmentin Panadol
B. null Panadol
C. Augmentin null
D. Error

arr[1] arr[0] Item Augmentin

8. What is the output of following Java program

```
public class Test
{ public static void main (String[]
args)
{
  Vitamins obj =new Vitamins
("Centrum",10,"poor Diet",2019);
  System.out.print(obj);
 }}
```

name Price usage year

toString

A. Something like Vitamins@2a139a55
B. Name:Centrum price: 10 Usage: poor Diet 2019
C. Name:Centrum
D. Name:Centrum price: 10 Usage: poor Diet

9. What is the output of following Java program?

```
public class Test
{public static void main (String[] args)
 {
Medicine Med =new Medicine
("Panadol",10,"headache");
Vitamins obj= new
Vitamins("Centrum",10,"poor Diet",2019);
obj.merge(Med);
System.out.print(Med.getName()+" "+
obj.getName()); }}
```

obj → cnt ~ panadol
null ~ centrum panadol

A. Panadol Centrum null
B. Panadol Centrum Panadol
C. null Centrum Panadol
D. Panadol Centrum Panadol

10. What is the output of following Java program?

```
public class Test
{ public static void main (String[]
args)
{
 Vitamins obj =new Vitamins
("Centrum",12,"poor Diet",2019);
 Medicine med = new Medicine
("Augmentin",100,"Flo");
obj.getDescription(med);
 }}
```

error, wrong type cast

A. Vitamins: 2019
B. medicine: 0
C. medicine: 2019
D. Error

11. What is the output of following Java program?

```
public class Test
{ public static void main (String[] args)
 {
 Item obj =new Vitamins
("Centrum",10,"poor Diet",2019);
 obj.print();
}}
```

centrum ~ 10 ~ poor diet 2019

A. Centrum 10 poor Diet 2019
B. Centrum 10 poor Diet
C. Centrum 10 poor Diet 0
D. Error

12. What is the output of following Java program?

```
public class Test
{ public static void main (String[]
args)
 {
  Vitamins obj =new Vitamins
("Centrum",10,"poor Diet",2019);
System.out.println(obj.computeDose(7)
); Medicine : & Centrum 10
 }}
```

A. medicine: Centrum 10
B. medicine: null 10
C. medicine: null 7
D. Error

Scanned with CamScanner

13. What is the output of following Java program?

```
public class Test
{public static void main (String[] args)
{
Item arr[] = new Item [3];
arr [0] =new Medicine
"Augmentin",100,"Flo");
arr [1] =new Medicine
("Panadol",10,"headache");
arr[2]= new Vitamins("Centrum",10,"poor
Diet",2019);
  for(int i = 0; i<3 ; i++)
   if (arr[i] instanceof Medicine)
System.out.print(arr[i].getName()+" ");
  }}
```

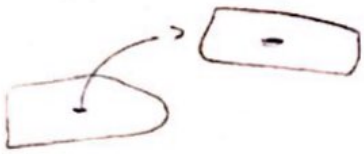
Aug ~ Panadol ~ centrum

A. Centrum
B. Augmentin Panadol
C. Augmentin Panadol Céntrum
D. Error

14. What is the output of following Jav

```
public class Test
{public static void main (Str
args)
{
Item arr[] = new Item [3];
arr [0] =new Medicine
("Augmentin",100,"Flo");
arr [1] =new Medicine
("Panadol",10,"headache");
arr[2]= new
Vitamins("Centrum",10,"poor
Diet",2019);
for(int i = 0; i<3 ; i++)
 if (arr[i] instanceof Vitamins
System.out.print(arr[i].getUsage
");
  }}
```

A. poor Diet
B. Flo headache
C. Flo headache poor Diet
D. Error



Scanned with CamScanner

Class Item and its Subclasses for Questions #1

```java
public abstract class Item
{
    private String name;
    public Item(String s)
    {
        name = s;
    }
    public String getName()
    {
        return name ;
    }
    public void setName(String n)
    {
        name=n ;
    }
    public String toString()
    {
        return "Name:"+name;
    }
    public abstract void print();
    public abstract int computeDose(int age);
}
```

```java
public class Medicine extends Item
{
    public int price;
    private String usage;
    public Medicine ()
    {
        this(null,0,null);
    }
    public Medicine(String name, int p, String usage)
    {
        super(name);
        this.price =p;
        this.usage=usage;
    }
    public void print()
    {
        System.out.print(super.getName()+" "+this.price+"
"+this.usage+" ");
    }
    public int computeDose(int age)
    {
        System.out.print("Medicine: " + super.getName()+" ");
        if (super.getName().charAt(0)=='U' && age >10)
            return 7;
        return 10;
    }
```

Scanned with CamScanner

```
    public String toString()
    {
        return super.toString()+" price: "+this.price+" usage: "
+this.getUsage();
    }
    public String getUsage()
    {
        return this.usage;
    }
    public void setUsage(String u)
    {
        this.usage=u;
    }
}
```

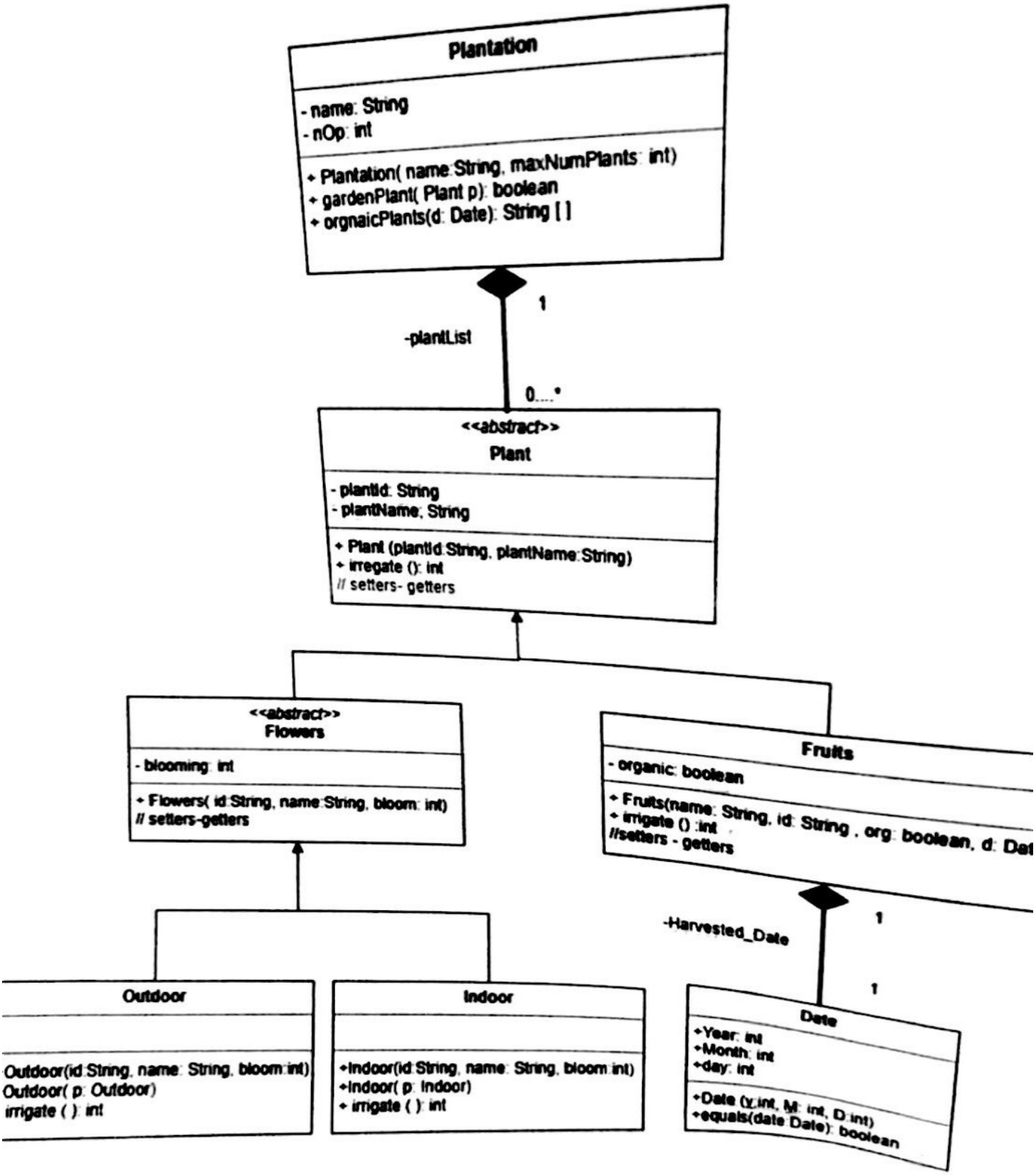
```
public class Vitamins extends Medicine
{
    private int year;
    public Vitamins(String name, int p, String usage, int w)
    {
        super(name,p,usage);
        this.year=w;
    }
    public void print()
    {
        super.print();
        System.out.print(this.year);
    }
    public void merge (Item item)
    {
        Medicine M =(Medicine) item;
        super.setName(super.getName()+" "+ M.getName());
        super.setUsage(super.getUsage()+" "+M.getUsage());
        M.setName(null);
        M.setUsage(null);
    }
    public void getDescription(Item item)
    {
    System.out.println(item.getClass().getName()+": "+
((Vitamins)item) .year);
    }
}
```

Augmentin , 100, 5TO

Medicine: ——

Vitamins موالها كانت Medicine

Scanned with CamScanner



Scanned with CamScanner

The description of UML:

Plant Class:

- Attributes:
  - *plantId:* plant id.
  - *plantName:* plant name.
- Methods:
  - *Plant(plantId: String , plantName: String):* constructor.
  - *irrigate( )*(ري): abstract method.

Fruits Class:

- Attributes:
  - *organic:* indicates whether the fruit is organic or not.
- Methods:
  - *Fruit(name: String , id: String, org: boolean, d:Date):* constructor.
  - *irrigate( ):* returns the number of time of irrigation for organic and non-organic fruits.

Flowers Class:

- Attributes:
  - *blooming:* the number of days the flower is bloomed.
- Methods:
  - *Flowers(id: String, name: String , bloom: int):* constructor.

Indoor Class:

- Methods:
  - *Indoor(id: String, name: String , bloom: int):* constructor.
  - *Indoor(p:Indoor):* constructor.
  - *irrigate( ):* returns the number of time that required for indoor plant irrigation.

Outdoor Class:

- Methods:
  - *Outdoor(id: String, name: Stri bloom: int):* constructor.
  - *Oudoor(p:Outdoor):* constructo
  - *irrigate( ):* returns the number o time that required for Outdoor pl irrigation.

Date Class:

- Methods:
  - *Date (y:int, M:int, D:int):*constru
  - *equals (date:Date):* this method returns true if the current date is same as the date sent by parameter

Plantation Class : (مشتل)

- Attributes:
  - *name:* plantation name
  - *nOp:* the number of plants in the plantation.
- Methods:
  - *Plantation(name: String, maxNumPlants: int):* constructor.
  - *gradenPlant(Plant p):* this method adds a plant (p) to the plantation if possible and returns true. Otherwise, the method will return false.
  - *OrganicPlants (d:Date):* this method returns an array containing **ONLY** the name of all organic fruits that have been harvested in (تحصد في) the specified date. ***Hint:*** *null elements are not allowed in the returned array.*

Scanned with CamScanner

1. **Write the class header of Indoor class**

```
Public class Indoor extends Flowers {
```

2. **Implement the constructor of class Fruits**

```
public Fruits (String name, String id, boolean org, Date d)
{
    super(name, id);
    organic = org ;
    Harvested_date = new Date (d.getYear(), d.getMonth(), d.getDay());

}
```

3. **Translate into Java code the following selected method from class plantation**

```
public boolean gardenPlant( Plant p){
{
    if(p instanceof fruits && nop < plantlist.(length){
    PList[nOP++] = new Fruit(p.getName(), p.getID(), p.getorg(), p.getDate());
    return true;
    }
    if(p instance of outdoor && nop < plantlist.length){
    PList[nOP++] = new outdoor(p);
    return true;
    }
    if(p instanceof indoor && nop < plantlist.length){
    PList[nOP++] = new Indoor(P);
    return true;
    }
    return false;

}
```

Scanned with CamScanner

```
public String [] organicPlants (Date date)
{
 String [] harvested list;
 int j=0;
 for(int i=0; i < nop; i++){
 if(plantlist[i] instance of fruits)
 if(plant list[i].getOrganic() && Plat list[i].getHarvested_date().equals(date))
   j++;
 harvested list=new string[j];
 for (int i=0, k=0; i < nop; i++){
} if (Plant list[i] instance of fruits)
 if (plant list[i].getOrganic() && Plant list[i].getharvested_date().equals(date){
 harvested list[k++] = plant list.getName();
 }
 }
 return harvested list;
}
```

4. Write Java code Statements for the application class (main method)

a. Create one Plantation called garden with the name "Al Mashtal" that contains a maximum of 10 plants.

```
Plantation garden = new Platation("Al Mashtal", 10);
```

b. Add indoor flower plant with the values (Id:"3224", name: "Orchid", blooming: 7) to the garden.

```
indoor flower("3224", Orchid, 7);
garden.gardenPlant(flower);
```

c. Print the name of all organic fruits in the garden that harvested in 2019-02-20.

```
String arr = garden.organicPlants(new Date(2019,2,20);
for (int i=0; i < arr.length; i++)
system.out.println(arr[i]);
```

Scanned with CamScanner