National University of Computer and Emerging Sciences

				Course Code:	SE2002
	Course:	Software Design & Architecture		Semester:	Spring 2022
	Program:	BS (SE) 4A		Total Marks:	20 16
	Duration:	60 Minutes (1 Hour)		Weight	15%
	Paper Date:	06-May-22		Page(s):	3
	Section:	All			
	Exam:	Sessional II			

Instruction/Notes:

Attempt all questions on the question paper. <u>Neither use nor submit any extra sheet.</u>

Name: _____   Roll Numb[_____]   Section ___

Question 1 (Max. Marks = 10)

9

associations?

Pizza making machines are of three types i.e. cheap pizza maker, moderate pizza maker, and expensive pizza maker. The cheap pizza maker makes a starch dough, a tomato filling, and a cheese topping. The moderate pizza maker makes a wheat flour dough, a minced meat filling, and a bell pepper topping. The expensive pizza maker makes a mixed grain dough, a meat slice filling, and an olive topping.

Map the information given above to a **UML 2 design class diagram** that uses the **Factory Method** design pattern. Your diagram should be annotated with relevant comments containing C++ code. Realistic and relevant assumptions may be made where necessary.
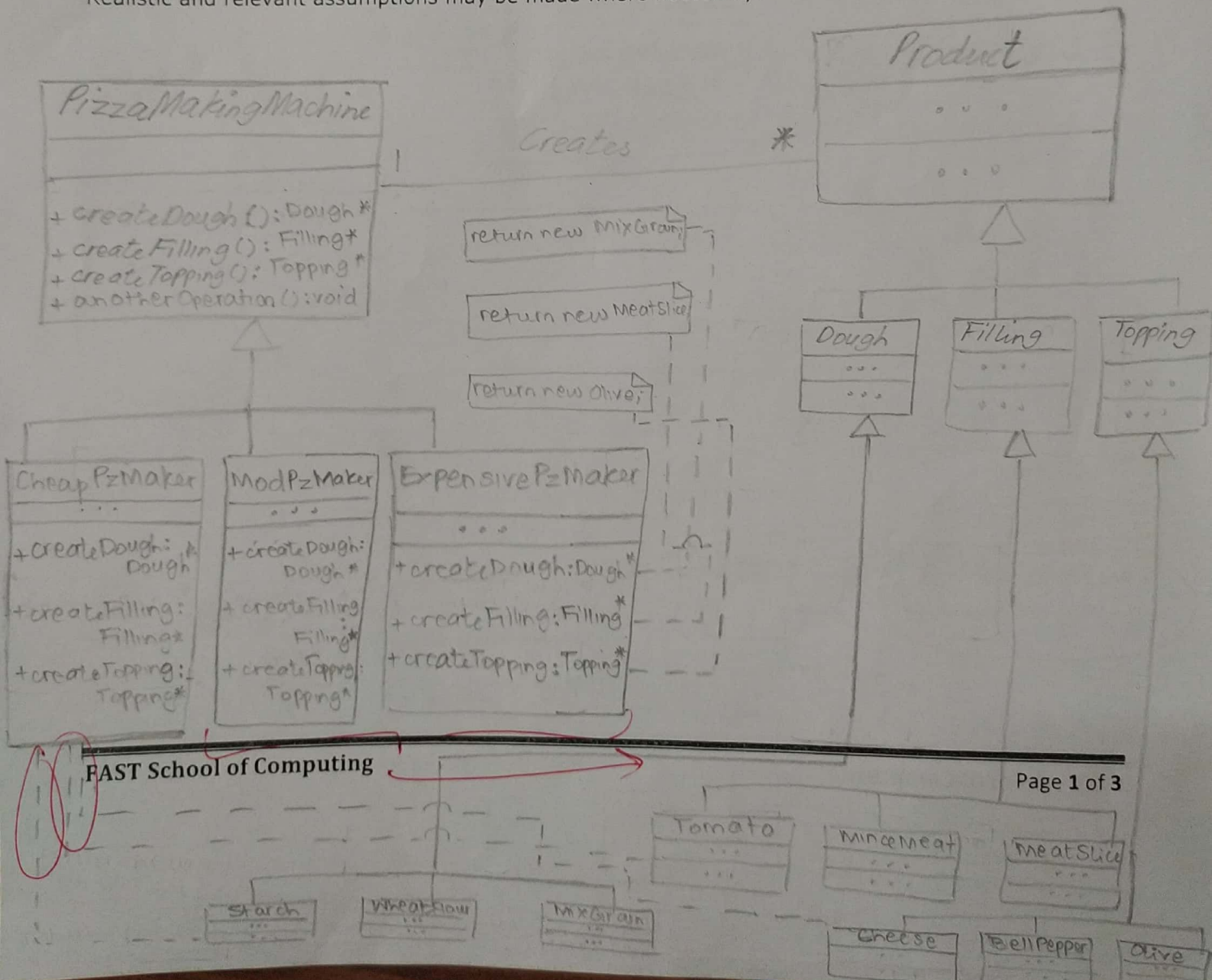
Question 2 (Max. Marks = 10)

7

Consider a simple file system. A node in the file system is either a file or a directory. A directory may contain one or more files. It may also contain one or more other directories. Every node has a name and size. The size of a directory is equal to the sum of the sizes of all the files and directories contained in it.

a. Which design pattern is applicable for the file system described above? [Note: An invalid answer to this part will result in a zero in the entire question.] **composite**

b. Draw a **UML 2 design class diagram** showing the design of the file system described above. This design **must** adhere to the structure of the design pattern chosen in part a above. Also, your diagram should be annotated with relevant comments containing C++ code. Realistic and relevant assumptions may be made where necessary.

Node

- name : string
- size : float

+ calculateSize():void
+ Add (F:File):void
+ Delete (F:File):void

1.. *

File

- size : float

+ calculateSize():void

Directory

- nodes : vector<Node*>
- size : float

+ calculateSize():void
+ Add (F:File):void
+ Delete (F:File):void

0..1

```
float newsize=0;
for(int i=0;i<F.size();i++)
{
    newsize += nodesli]->size;
}
```

```
nodes.pushback(&F)
```