ourdiu.com | Shared by Shanto Paul

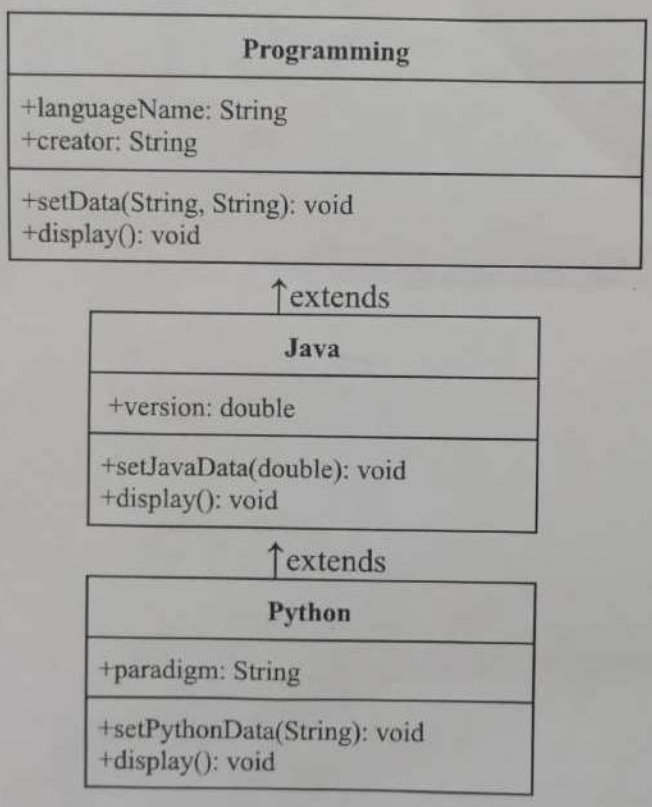
Daffodil International University
Faculty of Science & Information Technology
Department of Software Engineering
Midterm Examination, Spring 25
Course Code: SE216; Course Title: Object Oriented Programming
Sections: A-J & Teachers: RJM, MAAA, PC, NAN, KMH

Time: 1 Hour 30 Mins **Marks: 25**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1.	a)	**Convert** the below class diagram to the appropriate Java code to implement the class diagram. Create one object in Python class, implement all the methods and display the outputs. **Expected Output:** Basic Programming Java Programming Python Programming	[8]	CLO-1 *Level-2*
	b)	**Explain the** difference between static and non-static methods with examples.	[2]	
2.	a)	Utilize the given code below to write a **Spell class** (Design class) to get the output as shown. Here, you have to deal with constructor overloading and method overloading. **// Rewrite your design code here:**	[7]	CLO-2 *Level-2*



Driver Code:

```
public static void main(String[] args) {
    Spell expelliarmus = new Spell();
    Spell lumos = new Spell("Lumos", "Illuminating");
    Spell petrificus = new Spell("Petrificus Totalus");

    expelliarmus.printSpell();
    lumos.printSpell();
    petrificus.printSpell();
```

ourdiu.com | Shared by Shanto Paul

```
    expelliarmus.updateInfo("Stupefy");
    lumos.updateInfo(5);
    petrificus.updateInfo("Imperio", "Controlling");

    expelliarmus.printSpell();
    petrificus.printSpell();
}
```

Output:

Unknown spell is Mysterious.
Lumos spell is Illuminating.
Petrificus Totalus spell is Freezing.
Lumos spell requires 5 wands of power.
Stupefy spell is Mysterious.
Imperio spell is Controlling.

b) **Scenario:** [5]

Imagine you are building a simple class called **Person** for a personal information manager. Each Person object should store:

- A name (a string).
- An age (an integer).

However, you want to make sure that:

- The name and age cannot be modified directly from outside the class.
- There should be a way to set and get these values in a controlled manner.

Clarify the Person class and consider how you would structure the code so that the name and age are protected but can still be accessed and updated. Think about how you would allow others to interact with the properties while preventing them from directly changing the values.

[Note: You have to write full Java code here]

c) **Compute** the output of the following code: [3]

```
public class JigSaw{
        int instance=3;
        static double check= 10.5;
        public void puzzle0() {
                System.out.println(instance);
                System.out.println(check);  }
        public void puzzle1(){
                int rubics=50;
                check=15.5;
                instance=6;
                System.out.println(rubics);
                System.out.println(instance);
    }
}
public class Sudoku extends JigSaw {
        public void puzzle2() {
                System.out.println(instance);
                System.out.println(check);}
        public static void main(String[]atgs) {
                Sudoku solve1= new Sudoku();
                Sudoku solve2= new Sudoku();
                solve1.puzzle0();
                solve2.puzzle1();
                solve1.puzzle2();
        }
}
```

3
10.5
50
6
3
15.5