

Sample exam 2

The INFDEV team

1 Question 1

Given the following block of code, fill in the stack, heap, and PC with all the steps taken by the program at runtime.

- Points: 4 (50% of total).
- Grading: one point per correctly filled-in execution step.
- Associated learning objective: *abstraction*.

```
1 interface Animal {
2     void makeSound();
3 }
4 class Dog : Animal {
5     public Dog() {
6     }
7     public void makeSound() {
8         System.out.println("Woof!");
9     }
10 }
11 class Cat : Animal {
12     public Cat() {
13     }
14     public void makeSound() {
15         System.out.println("Miao!");
16     }
17 }
18 Animal myAnimal = new Cat();
19 myAnimal.makeSound();
```

1. Stack:

PC
1

2. Stack:

PC
18

Heap:

1
__type=Cat

3. Stack:

PC	...		PC	ret	this
18	...		13	null	ref 1

Heap:

1
__type=Cat

4. Stack:

PC	...		PC	ret
18	...		13	ref 1

Heap:

1
__type=Cat

5. Stack:

PC	myAnimal
19	ref 1

Heap:

1
__type=Cat

6. Stack:

PC	...		PC	ret	this
19	...		15	null	ref 1

Heap:

1
__type=Cat

7. Stack:

PC	...		PC	ret
19	...		15	null

Heap:

1
__type=Cat

Output: "Miao!"

8. Stack:

PC	myAnimal
20	ref 1

Heap:

1
__type=Cat

Output: "Miao!"

2 Question 2

Given the following block of code, fill in the declarations, class definitions, and PC with all steps taken by the compiler while type checking.

- Points: 4 (50% of total).
- Grading: one point per correctly filled-in type checking step.
- Associated learning objective: *type checking*.

```

1 class Employee {
2     private String name;
3     public Employee() {
4     }
5     public string GetName() {
6         return this.name;
7     }
8 }
9 class Manager : Employee {
10     public Manager() {
11     }
12     public string GetFunction() {
13         return "I am the boss!";

```

```
14     }
15 }
16 ...
17 Employee employee = new Manager();
```

1. Declarations:

PC
1

2. Declarations:

PC
8

Classes:

Employee
Employee=Employee → Employee GetName=Employee → string name=string

3. Declarations:

PC
15

Classes:

Employee	Manager
Employee=Employee → Employee GetName=Employee → string name=string	GetFunction=Manager → string GetName=Employee → string Manager=Manager → Manager name=string

4. Declarations:

PC
17

Classes:

Employee	Manager
Employee=Employee → Employee GetName=Employee → string name=string	GetFunction=Manager → string GetName=Employee → string Manager=Manager → Manager name=string

5. Declarations:

PC	employee
18	Employee

Classes:

Employee	Manager
Employee=Employee → Employee GetName=Employee → string name=string	GetFunction=Manager → string GetName=Employee → string Manager=Manager → Manager name=string