

COMP 2522 - Winter 2024 - Quizz 2

Note : I put one page per question but the answer can be very short for some of them.

1. Is there an error at the line 6 ? Why or why not ? (2 marks)

```
1  class Foo{
2      private Bar b;
3      private int x;
4      class Bar{
5          public void setX(int value){
6              x = value; // Error here
7          }
8      }
9  }
```

Answer :

2. Examine the main() method below to create a Bar object from the definition provided in Q1 above. The compiler say the line in main is in error. Why ? How do you fix it ? (2 marks)

```
1 public static void main(String[] args){  
2     Foo.bar bar = new Bar(); // Error  
3 }
```

Answer :

3. Is there an error in the code below ? Why or why not ? (2 marks)

```
1 interface A{
2     void foo();
3 }
4
5 class B implements A{
6     public void foo(){}
7 }
8
9 class Main{
10     public static void main(String[] args){
11         A a = new B(); // Error ?
12     }
13 }
```

Answer :

4. Using the below interface definitions and Q3, is there any conflict in class J ? (2 marks)

```
1 interface X{  
2     void foo();  
3 }  
4  
5 abstract class J implements A,X {}
```

Answer :

5. Modify the classes Animal, Dog and Cat as necessary and define CatDog to have Play functions as desired. Note Cat is-NOT a Dog is-NOT a Cat. Hint : Java does not allow multiple inheritance however it gives you something that allows a class to be many different types! Use that ability ! You cannot change class Play in any way. (3 marks)

```
1  class Animal{
2      public void draw(){
3          System.out.println("animal");
4      }
5  }
6
7  class Dog{
8      public void draw(){
9          System.out.println("dog");
10     }
11     public void bark(){
12         System.out.println("woof");
13     }
14 }
15
16 class Cat{
17     public void draw(){
18         System.out.println("cat");
19     }
20     public void purr(){
21         System.out.println("prrr");
22     }
23 }
24
25 class CatDog{}
26
27 // No changes needed below this line
28 public class Play{
29     public void guard(Dog d){
30         d.bark();
31     }
32     public void pet(Cat c){
33         c.purr();
34     }
35     public void make(Animal a){
36         a.draw();
37     }
38
39     public static void main(String[] args){
40         CatDog a = new CatDog();
41         Play p = new Play();
42         p.guard(a);
43         p.pet(a);
44         p.make(a);
45     }
46 }
47
48 // Desired output :
49 // woof
50 // prrr
51 // catdog
```

Answer :