Student id : Name : Date: 3/10/17

Quiz 2 - Lesson – 5 & 6 [25 Points] **Set - 1**

I. Say true or false. Just put T or F as an answer in the dash. (Each 1 Point)

1. super( ) can go anywhere within a subclass constructor. ___F____
2. A member inner class can access the private field and private method of its enclosing class (outer class). ___T_____
3. Java 8 default methods can override by the class which implements the interface. __T___
4. Reflection allows a program to inspect the contents of arbitrary objects at run time. ___T__
5. "IS-A" a relationship states that every object of the sub class is an object of the super class. This statement is _____T_____.
6. By default the visibility of a JFrame object is _____F___

II. Circle the right choice. (Each 1 Point)

7. What will be happen when the following code is compiled and the main method of the Main class is run? Circle the right choice.

```
interface Compute {
        default int compute(int x, int y) {
                return x + y;
        }
}
```

```
class Impl implements Compute{
}
public class QuizQuesion {
 public static void main(String args[]){
        Impl obj = new Impl();
        System.out.println(obj.compute(5, 3));
 }
}
```

A. Compiler error B. Runtime error C. Outputs 8 to the console

8. Which one of the given choice is related with user-input events in MVC – design Pattern?

a) Model b) View c) Controller d) None

9. In Swing, objects can add *listeners* to handle the events. The listeners are _____

a) Class b) abstract class c) interface

10. _____is a visual tool that comes with JavaFX.

a) JFC b) AWT c) SceneBuilder d) None

11. What is the output of the given program? Circle the right choice. **[1 Point]**

```
public class MemberClass {
        private String s = "Outer";
        public static void main(String[] args) {
                new MemberClass();
        }
        MyInnerClass myInner = new MyInnerClass();
        MemberClass() {
                        myInner.innerMethod();
                }
                private class MyInnerClass{
                        private String s = "Inner";
                        private void innerMethod(){
                          System.out.println(s);
                        }
                }
}
```

A. null B. Outer C. Inner D. Compilation Error

12. What is the output of the following code?

```
        interface i1 {
                default int show() {
                        return 100;
                }
        }
        abstract class A {
                public int show() {
                        return 50;
                }
        }
        public class TestClass extends A implements i1 {
                public static void main(String args[]) {
                        TestClass t = new TestClass();
                        System.out.println(t.show());
                }
        }
```

a) 100 b) 50 c) Run time Error d) Compile time error

13. Does the code have any error? Answer should be Yes or No.

Answer : ____YES___

```
class Outer {
   static int a=30;
   private int y = 100;
  static class Inner{
        static int x=50;
        static void display(){
              System.out.println("X =  "+x);
              System.out.println("A = "+a);
               System.out.println("Y = "+ y);
       }
   }
}
```

III. Answer the following questions

14. Find which line produces the error and write the reason why? Then fix the error. [3 Points]

abstract class AProduct{
private String productid;
public void setProductId (String id){
productid=id;
}
public String getProductId (){
return productid;
}
public abstract double getPrice();
}
	class Bicycle extends AProduct
	{
	public double getPrice(){
	return 230.45;
	}
	}
	class Test {
	public static void main(String[] args) {
	AProduct myProduct = new AProduct();
	Bicycle myCycle = new Bicycle();
	}

a. Write which line produces an error in the above code.
   AProduct myProduct = new AProduct();

b. Write the reason for the Error :
   Can't create object for Abstract Class

c. How do you fix this error? Write your Solution.

Solution 1 :

```
AProduct myProduct = new Bicycle();
```

Solution 2 :

```
Bicycle myProduct = new Bicycle();
```

15. What will the output of this given code? **[5 Points]**

```
public class Person {
        String fname;
        String lname;
        public Person(String fname, String lname) {
                this.fname = fname;
                this.lname = lname;
        }
}
class PersonWithJob extends Person {
        private int age;
        PersonWithJob(String fname, String lname,int age) {
                super(fname,lname);
                this.age = age;
        }
        @Override
        public boolean equals(Object ob) {
                if(ob == null) return false;
                if(this.getClass() != ob.getClass()) return false;
                PersonWithJob p = (PersonWithJob)ob;
                boolean isEqual= fname.equals(p.fname) && fname.equals(p.fname) &&
                                this.age == p.age;
                return isEqual;
        }
}
public class TestPersonEqual {
        public static void main(String[] args) {
                Person p1 = new Person("Renuka","Mohnaraj");
                Person p2 = new Person("Renuka","Mohnaraj");
                System.out.println(p1.equals(p2));
                System.out.println(p1.hashCode()==p2.hashCode());
                PersonWithJob p3 = new PersonWithJob("John","David",25);
                Person p4 = new PersonWithJob("John","David",25);
                System.out.println(p3.equals(p4));
                System.out.println(p4.equals(p3));
                Person p5 = null;
                System.out.println(p3.equals(p5));
```

```
}
}
```

Once you run TestPersonEqual class, what will be the output?

Answer :
False
False
True
True
False

16. Comparison of Actual Classes, Abstract Classes and Interfaces with the given specific Property**? Just fill the table cells with words "Allowed" or "Not Allowed". (Please don't use other words) [4 Points]**

Aspects	Actual Class	Abstract class	Java 8 Interface
Inclusion of constructor(s)	Allowed	Allowed	Not Allowed
Instances (Objects) of these aspects can be created directly is allowed or not.	Allowed	Not Allowed	Not Allowed
Inclusion of abstract methods	Not Allowed	Allowed	Allowed
Inclusion of static methods	Allowed	Allowed	Allowed