Started on	Wednesday, 20 January 2021, 9:10 PM
State	Finished
Completed on	Wednesday, 20 January 2021, 10:04 PM
Time taken	53 mins 41 secs
Marks	12.33/13.00
Grade	**9.49** out of 10.00 (**95**%)

Question **1**

Correct

Mark 1.00 out of 1.00

When the constructor of a class is called?

Select one or more:

- [ ] a. When defining the class (i.e. writing the UML diagram)
- [x] b. When creating an object ✔
- [x] c. When creating an instance of the class ✔
- [ ] d. When implementing the class (i.e. coding the class)

Your answer is correct.

The constructor is called when a new object is created (i.e. we are creating an instance of a class)

The correct answers are: When creating an instance of the class, When creating an object

Question **2**

Correct

Mark 1.00 out of 1.00

What are the elements of the interface of a method?

Select one or more:

- [x] a. return data type ✔
- [x] b. list of methods called by the method ✖
- [x] c. name ✔
- [ ] d. GUI
- [x] e. input parameters ✔
- [ ] f. Implementation

Your answer is correct.

- return data type of the method
- name of the method
- input parameters of the method

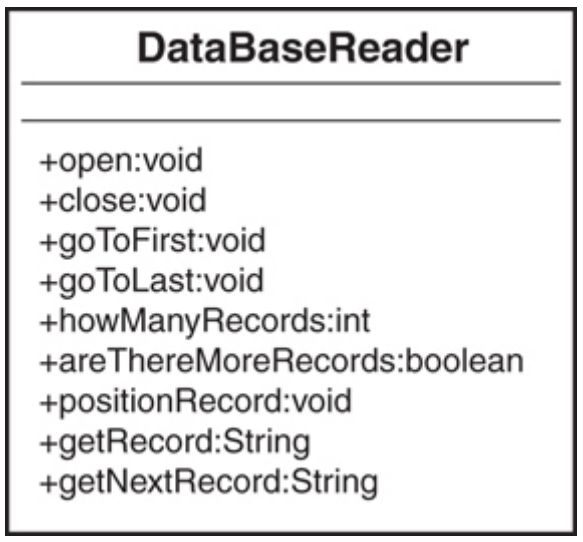
The correct answers are: return data type, name , input parameters

Question **3**

Correct

Mark 1.00 out of 1.00

In the following class diagram, which functionality is missing?



Select one or more:

- a. Position the cursor in the first record
- b. Add a record to the database ✔
- c. Retrieve a record from the database
- d. Calculate the number of records in the database

Your answer is correct.

There is no method to add a record to the database

The correct answer is: Add a record to the database

Question **4**

Correct

Mark 1.00 out of 1.00

The concept of method overloading can only be applied to the constructors of a class

Select one:

- True
- False ✔

Overloading allows a programmer to use the same method name over and over, as long as the signature of the method is different each time. This is applicable to constructors or any other method in the class.

The correct answer is 'False'.

Question **5**

Correct

Mark 1.00 out of 1.00

In every programming language, the return type of a method is part of the signature of the method

Select one:

- True
- False ✔

Depending on the language, the signature **may or may** not include the return type. In Java and C#, the return type is not part of the signature.

Do not mix the concept "*signature of a method*" with "*interface of the method*".

The correct answer is 'False'.

Question **6**

Correct

Mark 1.00 out of 1.00

Given the following implementation, how can we create a new object in Java?

```
public class DataBaseReader {

    String dbName;
    int startPosition;

    // initialize just the name
    public DataBaseReader (String name){
        dbName = name;
        startPosition = 0;
    };

    // initialize the name and the position
    public DataBaseReader (String name, int pos){
        dbName = name;
        startPosition = pos;
    };

    .. // rest of class. No more constructors
}
```

Select one or more:

- [ ] a.
```
DataBaseReader d1 = DataBaseReader ("database");
```

- [ ] b.
```
DataBaseReader d1 = new DataBaseReader ();
```

- [x] c.
```
DataBaseReader d1 = new DataBaseReader ("database");
```

Your answer is correct.

As soon as we define a constructor, the default constructor is not provided anymore.
In Java, we need to use the keyword "**new**" to create an object.

The correct answer is:
```
DataBaseReader d1 = new DataBaseReader ("database");
```

Question **7**

Partially correct

Mark 0.33 out of 1.00

Match the words with the explanations

It is declared inside the class, but outside all the methods. Each object has its own copy	Class Attribute ✗
It is declared inside the class, but outside all the methods. The keyword static was used in the declaration	Object Attribute ✗
It is defined inside a method.	Local Attribute ✓

Your answer is partially correct.

You have correctly selected 1.
The correct answer is: It is declared inside the class, but outside all the methods. Each object has its own copy → Object Attribute, It is declared inside the class, but outside all the methods. The keyword static was used in the declaration → Class Attribute, It is defined inside a method. → Local Attribute

Question **8**

Correct

Mark 1.00 out of 1.00

Java does not allow operator overloading.

Select one:

- True ✔
- False

The correct answer is 'True'.

Question **9**

Correct

Mark 1.00 out of 1.00

A synonym of "instance variable" is:

Select one:

- a. Local attribute
- b. Class attribute
- c. Object attribute ✔

Your answer is correct.

The correct answer is: Object attribute

Question **10**

Correct

Mark 1.00 out of 1.00

Is the following example correct?

```
public Time() {
    new Time();
    this.hour = 0;
    this.minute = 0;
    this.second = 0.0;
}
```

Select one:

- True
- False ✔

You **cannot** use the keyword "new" as in the example. Doing so causes an infinite recursion since new invokes the same constructor, which uses new again, which invokes the constructor again, and so on.

The correct answer is 'False'.

Question **11**

Correct

Mark 1.00 out of 1.00

Assume the following class:

```
public class Time {
   private int hour;
   private int minute = 0;
   private double second;
   public Time() {
        hour = 10;
        minute = 10;
        second = 10.0;
  }
   public Time(int hour, int minute, double second) {
       this.hour = hour;
       minute = minute;
       this.second = second;
   }
}
```

If we create a new object of the Time class as follows:

```
Time time1 = new Time();
```

What is the value of the instance variable "minute"?

Answer: 10 ✔

The correct answer is: 10

Question **12**

Correct

Mark 1.00 out of 1.00

Assume the following class:

```
public class Time {
   private int hour;
   private int minute = 0;
   private double second;
   public Time() {
        hour = 10;
        minute = 10;
        second = 10.0;
  }
   public Time(int hour, int minute, double second) {
       this.hour = hour;
       minute = minute;
       this.second = second;
   }
}
```

If we create a new object of the Time class as follows:

```
Time time1 = new Time(10,10,10.0);
```

What is the value of the instance variable "minute"?

Answer: 0 ✔

The names of the parameters are the same as the instance variables of the class. The input parameters hide the instance variables. This situation is called shadowing, because the parameter “hides” the instance variable with the same name.

To access the instance variable we need to use the keyword "this"

The correct answer is: 0

Question **13**

Correct

Mark 1.00 out of 1.00

Assume a **Time** class is defined and you created the following objects:

```
Time time1 = new Time(9, 30, 0.0);
Time time2 = time1;
Time time3 = new Time(9, 30, 0.0);
```

The output of the following code snippet will be "true":

```
if (time1 == time3 && time1==time2){
     System.out.println("true");
}else{
     System.out.println("false");
}
```

Select one:

- True
- False ✔

The correct answer is false. Review the section "The equals Method" on Chapter 11 on Think Java. I will elaborate on this topic during the upcoming lecture.

The correct answer is 'False'.

◄ Reading Quiz 1 - Sunday 17th January

Jump to...

Lecture Slides ►