

## Quiz Submissions - Quiz #9



Hoang Phuong Truong (username: htruong26)

### Attempt 1

Written: Mar 21, 2020 9:01 AM - Mar 21, 2020 9:20 AM

### Submission View

You successfully submitted your quiz.

### Question 1

1 / 1 point

Which principle of SOLID does the following code break?

```
public class Person
{
    private Long personId;
    private String firstName;
    private String lastName;
    private String age;
    private String accountNumber;
    private String accountName;
}
public class Account
{
    private Long guid;
    private String status;
    private String type;
}
```

- ☒ a) Single Responsibility Principle
- ☐ b) Open Closed Principle
- ☐ c) Liskov's Substitution Principle
- ☐ d) Dependency Inversion Principle

### Question 2

1 / 1 point

Application specific interfaces are better than general purpose interface is

- ☐ a) Single Responsibility Principle

- ☐ b) Open/ Close Principle
- ☐ c) Liskov's Substitution Principle
- ☒ d) Interface Segregation Principle

### Question 3

1 / 1 point

What is the singleton design pattern in Java?

have only one instance of the class

The correct answer is not displayed for Written Response type questions.

▼ Hide Feedback

Singleton pattern in Java is a pattern which allows only one instance of Singleton class available in the whole application

### Question 4

0 / 1 point

A class having high maintainability will tend to have which of the following characteristics

- ☐ a) High testability, low cohesiveness
- ☒ b) High testability, High cohesiveness
- ☒ c) Low testability, high cohesiveness
- ☐ d) Low testability, low cohesiveness

Consider the code below to answer the following two questions

```
package ca.bcit.quiz9A;  
  
public class Cat {  
    public Cat(){  
        System.out.println("My name is Fluffy");  
    }  
}
```

```

package ca.bcit.quiz9B;

public class Cat {
    public Cat(){
        System.out.println("my name is Kitty");
    } }

```

```

package ca.bcit.quiz9.main;
import ca.bcit.quiz9A.*;
import ca.bcit.quiz9B.*;
public class Driver {
    public static void main(String[] args) {
        Cat c = new Cat();
    } }

```

## Question 5

0 / 1 point

The Driver class will not compile, why?

Answer: compiler dont know which Cat class to be used to create an object

✗ (The class name is ambiguous to the compiler. Java does not know Which Cat class is being used)

## Question 6

1 / 1 point

What changes can be done to get the above code display "my name is Kitty"?

Answer: import ca.bcit.quiz9B.Cat ✗ (Use the full qualified name of the class Cat in package ca.bcit.quiz7B ca.bcit.quiz7B.Cat c = new ca.bcit.quiz7B.Cat();)

## Question 7

1 / 1 point

Which of the following principles is related with polymorphism concept?

- ✓ ☒ a) Liskov substitution Principle
- ☐ b) single responsibility Principle
- ☐ c) Open-closed Principle

☐ d) Interface Segregation Principle

Consider this class in answering the following two questions:

```
public class Book {
    private String title;
    private String authorName;

    public Book(String title,String authorName) {
        setTitle(title);
        setAuthorName(authorName);
    }
    public void setTitle(String title){
        if(title != null){
            this.title = title;
        } else {
            throw new IllegalArgumentException("title cannot be null");
        }
    }

    public void setAuthorName(String name){
        if(name != null){
            authorName = name;
        } else {
            throw new IllegalArgumentException("author name cannot be
null");
        }
    }

    public String getTitle(){ return title;}

    public String getAuthorName (){ return authorName;}

    public void printCurrentPage(){
        // code to send the current page the printer
    }
}
```

### Question 8

0 / 2 points

This code breaks one of the SOLID code principles. Which principle does it break, explain?

Open/Closed Principle

The correct answer is not displayed for Written Response type questions.

▼ [Hide Feedback](#)

Single Responsibility Principle SRP, the method print current page should not be a part of the Book class and its not one of the responsibilities of a Book object

### Question 9

0 / 1 point

How can the code above be refactored so that it will not break any SOLID code principles?

- No text entered -

The correct answer is not displayed for Written Response type questions.

▼ Hide Feedback

Move method printCurrentPage() to another class such as Printer class

### Question 10

2 / 2 points

Mention two possible results of a duplicated code?

poor maintainability  
Expensive to fix bugs

The correct answer is not displayed for Written Response type questions.

▼ Hide Feedback

Poor maintainability, hard to fix bugs and hard to keep up with changes.

---

Attempt Score: 7 / 12

Overall Grade (highest attempt): 7 / 12

Done