Quiz Submissions - Quiz #2

Hoang Phuong Truong (username: htruong26)

Attempt 1

Written: Feb 1, 2020 9:15 AM - Feb 1, 2020 9:36 AM

Submission View

You successfully submitted your quiz.

Question 1 **1 / 1 point**

Which Scanner class method is used to read an integer from the user?

- readInt()
- nextInteger()
- next()
- ✔ nextInt()

Question 2 **0 / 2 points**

A novice programmer was asked to write a method to count and display the number of tokens of user's input, below is the method the programmer wrote:

```
import java.util.Scanner;
public class Quiz2 {
public static void getCount(){
Scanner input = new Scanner(System.in);
int count = 0;
System.out.println("enter some text: ");
while(input.hasNext()){
count++;
System.out.println(count);
}
}

public static void main(String[] args){
Quiz2.getCount();
```

```
}
}
1) Will the above code compile?
2) If the code compiled, what will the output be for the input
(java is an object oriented language) ?
```

Answer for blank # 1: No ✗

Answer for blank # 2: 6 ✗

(Yes, the output will be an infinite loop because the tokens are not being retrieved from the buffer using a method like next() or nextLine())

Question 3 **1 / 1 point**

_____ is a measure of the degree of independence between classes

- ○ Cohesion
- ✓ ○ Coupling
- ○ All of the above
- ○ None of the above

Question 4 **1 / 1 point**

Which method does not store the String value after the space?

- ✓ ○ 1) next()
- ○ 2) nextString()
- ○ 3) nextLine()
- ○ 4) Both 1 and 3

Question 5 **1 / 4 points**

1. Take a look at the following classes and answer the following questions

```
package ca.bcit.quiz2;
public class Student {
```

```
   String name;
   int courseMark;
public Student(String name, int courseMark) {
     setName(name);
     setCourseMark(courseMark);
}
public void setName(String name){
  if (name != null && name.length() > 0){
     this.name = name;
  } else{
  throw new IllegalArgumentException("name cannot be null");

  }}

public void  setCourseMark(int mark){
 if(mark >= 0 && mark <= 100){
  coursrMark = mark;
}else{
throw new IllegalArgumentException("course mark should be between
0 and 100 ");
}
} // method setCourseMark end

}// class student end

package ca.bcit.quiz2;

import java.util.ArrayList;
public class Course {
private String courseName;
private ArrayList<Student> students;

public Course(String courseName) {
     this.courseName = courseName;
     students = new ArrayList<Student>();
}

public void displayStudentName(Student s){

System.out.println(" the student name is "+s.name );
}
/**
 * method used to update students
 * 1 for add student to the course
```

```
 * 2 for remove student from the course
 * 3 for edit student mark
 * @param updateType
 * @param s
 * @param newMark
 */
public void updateStudent(int updateType,Student s ,int newMark){
     if(updateType == 1){
          students.add(s);
     }else if (updateType == 2) {
          students.remove(s);
     }else if(updateType == 3){
          s.courseMark = newMark;
     }else{
          System.out.println("invalid update type");
     }}
```

1) Are these classes tightly coupled or loosely coupled?
2) What can be done to class **Student** to improve the design of the project?
3) Are these classes highly cohesive?
4) what can be done to class **Course** to improve the design of the project?

Answer for blank # 1:	loosely coupled	✖
Answer for blank # 2:	create mutator for field "name" with validation and create accessors for all fields	✖
Answer for blank # 3:	Yes, the classes are highly cohesive since focusing on a specific task	✖
Answer for blank # 4:	I would create separated methods to add student, remove student and updateMark with validation. Then in updateStudent, according to updateType, it will call an appropriate method.	✖

(1) The classes are tightly coupled because all the instance variables can be directly accessed from class Course . Because no visibility modifier was specified the default one (private package) would allow class Course to access the fields of class Student directly, 2)Make all the instance variables private and use accessors and mutators to access the fields from another class, 3) Class Course in not highly cohesive because it has method displayStudentFullName()

which should be in Student class and method updateStudent() does several tasks, 4) Move method displayStudentFullName() to Student class and break method updateStudent to several methods such as addStudent() removeStudent() editStudentMark())

Question 6

1 / 1 point

Consider the following code segment

```
public static void missingChar(){
String[] alphabet = {"a","b","c","d","e","f","g","h","i","g","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"};
Random rand = new Random();
// missing statement
System.out.println(alphabet[k]);
}
```

It is supposed to print a randomly chosen letter of the alphabet. Any of the 26 letters can be chosen with equal probability.Assuming that Random class was properly imported, which of the following can replace the // missing statement

- ✔ int k = rand.nextInt(26);
- int k = rand.nextInt(25);
- int k = rand.nextInt(26) + 1;
- int k = rand.nextInt(25) + 1;

Question 7

2 / 2 points

A software developer must design the classes with the goal of ___high___ ✔(50 %) cohesion and ___low___ ✔(50 %) coupling

---

Attempt Score:7 / 12

Overall Grade (highest attempt)**:**7 / 12

Done