CS2110 Exam 1 – Spring 2017

SOLUTIONS

Question 1 [**4 points**]: List any four (4) errors in the following code.

Answer: 1. Line 3, myArray is an array of Strings not ints. 2. Line 3, there are 6 elements in the right of the assignment but myArray was created to hold 5 elements. 3. Line 3, no semi colon to complete the statement. 4. Line 4, int i is not initialized. 5. Line 5, word is a String so we should use .equals() not ==. 6. Doesn't return anything it the condition in the if statement isn't true.

Question 2 [**8 points**]:

	Console	Your Response (true or false)
A	**CS 2110**	**true**
B	0.5	**False**
C	4242	**True**
D	10.0	**True**
E	7>8	**False**
F	180	**False**
G	b5	**False**
H	A	**False**
I	{1, 2, 3}	**False**

Question 3 [**3 points**]:

Answer: b A collection of elements that does not contain duplicates and is unordered

Question 4 [**3 points**]:

Answer: byte, short, int, float, long, double
(Note that the **order of int and float doesn't matter** and the **order of long and double doesn't matter**)

Question 5 [**6 points**]:

```
  public static boolean foo (int[] list){
    boolean isSet = true;
    for(int i = 0; i < list.length; i++){
        for(int j = i + 1; j < list.length; j++){
            if(int[i] == int[j]){
                isSet = false;
            }
        }
    }
    if(isSet){
        System.out.println("This is a set");
        return true;
    }else{
        System.out.println("This is not a set");
        return list.length < 10;
    }
}
```

Question 6 [**7 points**]:

Solution:

Dinosaur
- isHerivore : boolean - numberOfLegs : int - weight : double
+ Dinosaur(weight : double, numberOfLegs : int, isHerivore: boolean) + eat() : void + sleep(duration : int): boolean + roar() : void

1 pt for "Dinosaur" Class name
2 pts for three fields
1 pt each for 3 methods and 1 constructor (total 4 pts)

Question 7 [**3 points**]: Explain what an overloaded constructor does and why a programmer might want to have a class with more than one constructor.

Constructor overloading is a way to have more than one constructor, which does different tasks. Each constructor has a different method signature. [2 pts]

Typically a programmer would want more than one constructor so that at the time of creating an instance of an object, the programmer can choose which task is most appropriate by passing in more or less parameters when calling the constructor. [1 pt]

-or-

If at the time of creating an instance of an object, different information may be known about that object. So you can choose a constructor that performs a particular task that has the appropriate number and types of parameters to suit the current needs.

Question 8 [4 points]:

Phase	**Approximate Budget**	**Adjusted Budget**
Requirements	$200	
Specification	$500	
Design	$600	
Module coding	$6700	e.g. $500
Module testing	$700	
Integration	$800	
Maintenance	$500	e.g. $6700

Adjust the budget in the third column (above) if you think it's a poor plan and explain why. Otherwise give justification as to why you think this is a good plan.

[2 points] for decreasing the budget for "Module coding" and increasing the budget for "Maintenance" (Exact dollar amount is not important as the ratios.)

[2 points] reason: maintenance is the most expensive phase in relation to all other phases. Good software systems are maintained and any change to the software results in going through small software development life cycles each time, even within the maintenance phase.

Question 9 [4 points]: What is printed to the console after the following code snippet is executed?

Answer:

14

(Partial credit could be earned if work is shown)

Question 10 [9 points]:

[1.5 points each]

1) true [✓ false]

2) [✓ true] false

3) true [✓ false]

4) [✓ true] false

5) [✓ true] false

6) true [✓ false]

Question 11:

(A) [**9 points**]

```
@Test
public  void  testMapValues() {

     // SET UP
     // create a TreeMap and construct the scenario where two keys
     // are associated with the same value
     TreeMap<String, Integer> myMap = new TreeMap<String, Integer>();
     myMap.put(“Virginia”, 25);
     myMap.put(“California”, 25);

     QNineClass tester = new QNineClass();

     // ASSERT STATEMENT
     assertTrue(tester.mapValues(myMap));

}
```

Point distribution:

@Test	1pt
public	1pt
void	1pt
appropriate method name with ‘test’ in it	1pt
set up	3pts
assert statement	2pts

(B) [**2 points**]

Black-box test

Question 12 [8 points]:

```
// assuming the Cat class has the fields furColor and age
```

```
public boolean equals(Object o) {

     if(o instanceof Cat) {         // is o a Cat?
          Cat c = (Cat) o;          // cast o to Cat
          // are furColor and age equal?
          return c.furColor.equals(this.furColor) &&
                 c.age == this.age;
     }
     return false;

}
```

public	0.5 pts	
boolean	0.5 pts	
equals(Object o)	1 pt	
instanceof	1 pt	
casting	1 pt	
.equals for fur	2 pts	(because fur is String)
== for age	1 pt	(== is ok since age is an int)
appropriate returns	1 pt	

Total: 8 points

This is the end of this exam. Remember to sign the pledge, put your ID on every page, and bubble in your ID on the first page!