Exam #2 Review Questions
CSE110 - Arizona State University

1. Which of the following enforces encapsulation?

   a. Make instance variables `private`

   b. Make methods `public`

   c. Make the class `final`

   d. Both a and b

   e. All of the above
   The answer is d.

2. Use the following class to answer the questions below:

```
public class Store {
     private int quantity;
     private double price;
     public Store (int q, double p) {
          quantity = q;
          price = p;
     }
     public int getQuantity() {
          return quantity;
     }
     public void setPrice(double p) {
          price = p;
     }
     public double calcTotal() {
          return price * quantity;
     }
}
```

   a. What is the name of the class?
   `Store`

   b. List all instance variables of the class.

```
private int quantity
private double price
```

   c. List all methods of the class.

```
public Store (int q, double p)
public int getQuantity()
public void setPrice(double p)
public double calcTotal()
```

   d. List all mutators in the class.

```
public void setPrice(double p)
```

e. List all accessors in the class.
```
public int getQuantity()
```

f. List which method is the constructor
```
public Store(int q, double p)
```

g. Write the mutator for the quantity.
```
public void setQuantity(int q) {
     quantity = q;
}
```

h. Write the accessor for the price.
```
public int getPrice() {
     return Price;
}
```

i. Write the line of code that will create a new instance called videoStore that has 100 quantity and a price of 5.99.
```
Store videoStore = new Store(100, 5.99);
```

j. Call the calcTotal method with the videoStore object in part i to print out the total.
```
System.out.println(videoStore.calcTotal());
```

3. True or false? If no constructor is provided then Java automatically provides a default constructor. True. If there is no constructor defined by the programmer, Java will consider a default constructor.

4. True or false? A method must have at least one return statement. False, we can have a method with return type of void, so there is no need to have a return statement.

5. Correct the following class definition if you think it will not work:

```
public class Student {
     private String name;
     private String major;
     public Student() {
          name = "???";
          major = "xxx";
     }
     public Student(String n, String m) {
          n = name;
          m = major;
     }
     public String getMajor() {
          return m;
     }
     public String getName() {
          return n;
     }
}
```

The correct class form is as below:

```
public class Student {
     private String name;
     private String major;
     public Student() {
          name = "???";
```

```
        major = "xxx";
    }
    public Student(String n, String m) {
        name = n;
        major = m;
    }
    public String getMajor() {
        return major;
    }
    public String getName() {
        return name;
    }
}
```

6. What will be the output of the following loops? Indicate the number of times the output will be displayed if it is too many to list.

   a.
   ```
   int n = 979;
   for (int j = 0; j <= n; j++ ) {
       System.out.print("Hello ");
   }
   ```
   It will print out 980 "Hello" in one line

   b.
   ```
   int n = 5;
   for (int j = 1; j <= n; j+=3) {
       System.out.print("Hello ");
       int k=j;
       while (k < n) {
           System.out.println("Good Morning");
           k++;
       }
       j--;
   }
   ```
   Hello Good Morning
   Good Morning
   Good Morning
   Good Morning
   Hello Good Morning
   Good Morning
   Hello

   c.
   ```
   int j = 1;
   int n = 5;
   while (j <= n) {
       System.out.print("Hello ");
       n--;
   }
   ```
   Hello Hello Hello Hello Hello

   d.
   ```
   int j = 1;
   while(j <= 11) {
       System.out.print("Hello ");
       j = j + 3;
   }
   ```
   Hello Hello Hello Hello

   e. What is the output of the following?

```
String name;
int i;
boolean startWord;
name = "Richard M. Nixon";
startWord = true;
for (i = 0; i < name.length(); i++) {
     if (startWord)
          System.out.println(name.charAt(i));
     if (name.charAt(i) == ' ')
          startWord = true;
     else
          startWord = false;
}
```

R
M
N

f. What is the value of n?

```
int n = 1, i = 1;
while (i < 7) {
     n = n * i;
     i += 2;
}
System.out.print(n);
```

15

7. Write a boolean method called allDifferent that takes three int numbers and returns true if the numbers are all different and false otherwise.

```
boolean allDifferent(int a, int b, int c){
    if (a!=b && a!=c && b!=c)
        return true;
    return false;
}
```

Another way to do it is to return in one line:

```
boolean allDifferent(int a, int b, int c){
    return a!=b && a!=c && b!=c;
}
```

8. Write a loop that read in int values until the user enters 0 and prints out how many values entered are greater than 10.

```
Scanner in = new Scanner(System.in);
int count = 0;
int num = in.nextInt();
while (num!=0){
    if (num>10)
        count++;
    num = in.nextInt();
}
System.out.println(count);
```

9. Write a loop that will print out every other letter in a String str. For example if the String was "Hello There" then "HloTee" would be printed.

```
for (int i=0;i<str.length();i+=2)
     System.out.print(str.charAt(i));
```

10. Implement a class named `AsuStudent`. The class should keep track of the student's name, number of classes registered, hours spent per week for a class (Consider a student devotes the same amount of time for each of his class in a week). Implement a `toString` method to show the name and number of classes registered by a student, a `getName` method to return the name of the student, a `getTotalhours` method to return the total number of hours spent by a student in a week, and a `setHours` method to set the number of hours the student devotes for each class.

```
public class AsuStudent {
    private String name;
    private int numClass;
    private int hourPerClass;

    public AsuStudent(String name, int numClass, int hourPerClass) {
        this.name = name;
        this.numClass = numClass;
        this.hourPerClass = hourPerClass;
        }

    public String toString(){
        return name + ", number of class: "+ numClass;
     }
    public String getName(){
        return name;
        }
    public int getTotalHours(){
        return numClass * hourPerClass;
        }
    public void setHours(int newHours){
        hourPerClass = newHours;
        }
}
```