# Quiz 2 Questions

## Question 1

What is the purpose of a constructor?

The purpose of a constructor is a block of code tha tis automatically executed when an object is constructed. It initializes the object's data to sensible initial values.

## Question 2

What is wrong with the following code? Explain it and fix it.

void Hello(String s)  
{  
    if((!s.isBlank()) && (s != null))  
    {  
        System.out.println("Hello " + s.toUpperCase());  
    }  
}

Hello(String s)  
{  
   if(s != null || !s.isBlank())  
   {  
       System.out.println("Hello " + s.toUpperCase());  
   }  
}

Notes:  
- do not need quotes around !s.isBlank()  
- You have to as a main to run the event  
- the != null must come before !s.isBlank()  
- replace && with || because it can be either null or blank  
- if Hello is a constructor you do not need the word void there

### Question 3

Write the code to print out the first commandline argument from main. Write the entire method.

public class TechnicalThing {  
   // go to run > edit configurations  
   // Then in program configurations you can type a bc def - without quotes  
   public static void main(final String[] args)  
   {  
       //grabs first word in the commandline argument  
       System.out.println(args[0]);  
   }  
}

### Question 4

Consider a method called processNotes(ArrayList<String> notes, double someNumber) that does all of the following:  
- If the ArrayList is not null (1 mark) and not empty (1 mark), then  
- use a for-each loop to iterate through the notes (1 mark)  
- in each iteration: if the note is not blank (1 mark) and not null (1 mark), print the note in lower case (1 mark), but only if it either starts or ends with the word "hello" (2 marks) or contains the word "goodbye" (1 mark) (these matches are all case-insensitive, so "HEllo" and "gooDBye" etc are also valid matches) (1 mark). Finally, at the end of the method, it should print the value of someNumber using printf, to three decimal places (1 mark)

public void processNotes(final ArrayList<String> notes, final double someNumber)  
{  
   if(notes != null && !notes.isEmpty())  
   {  
       for(String note:notes)  
       {  
           if(note != null && !note.isBlank())  
           {  
    **if(note.toLowerCase().startsWith("hello") ||**  
**note.toLowerCase().endsWith("hello") ||**  
**note.toLowerCase().contains("goodbye"))**  
               {  
                   System.out.println(note.toLowerCase());  
               }  
           }  
       }  
   }  
   System.out.printf("The value of someNumber is %.3f", someNumber);  
}

### Question 5

Write the Java class and main class that are equivalent to the following python class:

class Car:  
 def \_\_init\_\_(self, year, make, model):  
 self.make = make  
 self.model = model  
 self.year = year  
  
 def drive(self):  
 print("vroom")  
  
  
def main():  
 car = Car(1963, "corvette", "stingray")  
 print(str(car.year) + " " + car.make + " " + car.model)  
 car.drive()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

class Car {  
   private final int year;  
   private final String make;  
   private final String model;

   Car(final int year, final String make, final String model)  
   {  
       this.year = year;  
       this.make = make;  
       this.model = model;  
   }

   public int getYear()  
   {  
       return year;  
   }

   public String getMake()  
   {  
       return make;  
   }

   public String getModel()  
   {  
       return model;  
   }

   public void isDriving()  
   {  
       System.out.println("vroom");  
   }

   public static void main(final String[] args)  
   {  
       Car car;  
       car = new Car(1963, "corvette", "stingray");  
       System.out.println(car.getYear() + " " + car.getMake() + " " + car.getModel());  
       car.isDriving();  
   }  
}

### Question 6

Write a Java class named MovieTheatre that has an instance variable named movieTitles, which is an ArrayList of String references. The MovieTheatre constructor must:  
- instantiate the ArrayList  
- add the strings "Groundhog Day" and "Dead Poet's Society" to the ArrayList  
- use a foreach loop to print all of the titles   
- remove the first element in the ArrayList  
- print how many movies are in the ArrayList

- print the longest movie title (use loops and String methods to find it)  
- print the first movie title from the ArrayList.  
Include all the code needed.

import java.util.ArrayList;

class MovieTheatre {  
   private String movieTitle;  
   private final ArrayList<String> movieTheatres;

   MovieTheatre()  
   {  
       movieTheatres = new ArrayList<>();

       movieTheatres.add("Groundhog Day");  
       movieTheatres.add("Dead Poet's Society");

       System.out.println("--Print Size of ArrayList--");  
       System.out.println("The Size of the ArrayList is " + movieTheatres.size());

       //Printing all  
       System.out.println("--Print All--");  
       for(String movie: movieTheatres)  
       {  
           if(movie != null)  
           {  
               System.out.println(movie);  
           }  
       }

       System.out.println("--Print only first item in ArrayList--");  
       System.out.println(movieTheatres.get(0));

       System.out.println("--Print the longest movie title--");  
       String longestMovie = "";  
       for(String movie: movieTheatres)  
       {  
           if(movie != null)  
           {  
               if(movie.length() > longestMovie.length())  
               {  
                   longestMovie = movie;  
               }  
           }  
       }  
       System.out.println(longestMovie);

       System.out.println("--Remove first element in ArrayList--");  
       movieTheatres.remove(0);  
       for(String movie: movieTheatres)  
       {  
           if(movie != null)  
           {  
               System.out.println(movie);  
           }  
       }

       System.out.println("--Print Size of ArrayList--");  
       System.out.println("The Size of the ArrayList is " + movieTheatres.size());

       System.out.println("--Print only first item in ArrayList--");  
       System.out.println(movieTheatres.get(0));

   }

   public static void main(final String[] args)  
   {  
       MovieTheatre movies;  
       movies = new MovieTheatre();  
   }  
}

Feedback

declare as List

initialize to null, not ""

### Question 7 \*\*\* important

Write a Java class named Netflix that has an instance variable named movieTitles, which is an array of String references. The Netflix constructor must:  
- instantiate the array  
- add the strings "Groundhog Day" and "Dead Poet's Society" to the array  
- use a for loop (not a foreach) to print all of the titles  
- remove the first element in the array (i.e. set it to null)  
- print how many movies are in the array  
- print the first movie title from the array.

public class Netflix  
{  
   private final String[] movieTitles;

   public Netflix()  
   {  
       //instantiate the array  
       movieTitles = new String[2];

       //add the strings "Groundhog Day" and "Dead Poet's Society" to the array  
       movieTitles[0] = "Groundhog Day";  
       movieTitles[1] = "Dead Poet's Society";

       //use a for loop (not a foreach) to print all the titles  
       if(movieTitles != null)  
       {  
           for (int i = 0; i < movieTitles.length; i++)  
           {  
               if (movieTitles[i] != null)  
               {  
                   System.out.println(movieTitles[i]);  
               }  
           }  
       }

       //remove the first element in the array (i.e. set it to null)  
       movieTitles[0] = null;

       //print how many movies are in the array  
       System.out.println(movieTitles.length);

       //print the first movie title from the array  
       System.out.println(movieTitles[0]);  
   }  
}

### Question 8

Write a Java class named Amazon that has an instance variable named items, which is a HashMap of key-value pairs; the key is an Integer and the value is a String. The Amazon constructor must:  
- instantiate the HashMap  
- add the key-value pairs 1234->"blue guitar" and 5678->"yellow umbrella" to the HashMap  
- use a Set, the keySet() method, and an iterator, print all the keys and values in the form of "key is 1234 and value is blue guitar"

import java.util.HashMap;  
import java.util.Iterator;  
import java.util.Map;  
import java.util.Set;

public class Amazon  
{  
   private final Map<Integer, String> items;

   public Amazon()  
   {  
       //instantiate the HashMap  
       items = new HashMap<>();

       //add the key-value pairs to the HashMap  
       items.put(1234, "blue guitar");  
       items.put(5678, "yellow umbrella");

       //use a Set, the keySet() method, and an iterator  
       //print all the keys and values in certain format  
       Set<Integer> allKeys;  
       Iterator<Integer> it;  
       Integer key;  
       allKeys = items.keySet();  
       it = allKeys.iterator();

       while(it.hasNext())  
       {  
           key = it.next();  
           System.out.printf("key is %d and value is %s.\n", key, items.get(key));  
       }  
   }  
}