Preguntas

1. Paradigmas de programación en Java
   Los paradigmas mas populares son:
   - Estructurada
   - Orientada a Objetos
   - Lógica y Funcional
2. Componentes de POO
   - Variables de referencia:
     Es un nombre que usamos para identificar hacia donde apuntamos.
   - Referencia:
     Básicamente son Identificadores de instancias que se encuentran en una clase.
   - Objetos:
     Son espacios en memoria que contienen valores que hacen una representación de alguna cosa.
3. String vs StringBuilder:

STRING	STRINGBUILDER
Es INMUTABLE, lo que significa que, si nace con un valor, este ya no puede cambiar, hasta que el programa muera.	Es MUTABLE, Su valor puede ser alterado las veces que sean necesarias, una característica es que, si hay mas StringBuilder apuntando al mismo objeto, a todos cambiara su valor.
EQUALS-> Compara si tiene la misma cadena	Compara si esta apuntando al mismo objeto
Cada vez que modificamos un String se crea un nuevo objeto.	Cada que se modifica un StringBuilder el objeto sigue siendo el mismo mas no su valor.

4. Representa -> Una variable de referencia no tiene ninguna referencia o no apunta a ningún objeto.
5. Es un constructor que se crea en automático si nosotros como programadores no especificamos uno dentro de nuestra clase, si la clase reconoce que se ha creado 1 entonces en constructor default dejaría de existir.

6. Tipos de Variables.

Local	Instancia	Staticas
Solo pueden ser accedidas dentro del método donde son declaradas y de ninguna otra parte del proyecto.	Son variables que están definidas dentro de un objeto, pero no tienen un modificador de acceso Static si no suelen usar Public, Private, Protected, Cada que se crea un nuevo objeto también se van a crear una copia de todas las variables de instancia que contenga y estas serán independientes para cada uno.	Estas son variables de clase y para acceder a ellas no es necesario instanciarla y se puede acceder desde cualquier parte de la clase

Capitulo 2

A. Ejercicio 2
   Respuesta: D, F y G
   ¿Por qué?: Aunque las variables no se hayan inicializado, el programa se va a ejecutar correctamente sin ningún error y el resultado va a ser los valores por defecto para cada tipo de Dato, eso el byteCode de Java lo sabe.
B. Ejercicio 10
   Respuesta: Solo imprimirá -> null para la variable color y 0 para la variable age
   ¿Por qué?: En primera por que La variable color es de tipo String, recordemos que es INMUTABLE por consecuencia al no inicializarla el byteCode le asigna null, entonces, aunque nosotros tratáramos de modificar su valor de este no cambiara. Ahora con la variable age sucede algo similar se le asigno su valor por defecto, el cual en el código de ejemplo no podremos cambiar tampoco.

Capitulo 4

C. Ejercicio 6.
   Respuesta: El código no compila.

   ¿Por qué?: El método estático “addCandy” pide un valor de retorno que es “long” y al momento de tratar de hacer el retorno el casteo lo intenta con “int”, también si ahí mismo se esta haciendo la suma de 2 valores distintos se deberían colocar entre paréntesis, para que primero haga la suma y después el casteo.

D. Ejrcicio 14

   Respuesta:A,B

   ¿Por qué?: A: Los valores de retorno no pueden ser nulos para tipos primitivos, hay una excepcion con los objetos, ahí el valor de retorno puede ser nulo.

   B: El operador de desigualdad comparar si se esta apuntando a distintos objeto s con el tipo de dato StringBuider, en el caso de Stirng comparara si las cadenas son distintas.