**Alumno: Ana Paulina López Cazares Fecha: 10 de Sep del 2019**

**Materia: Programación Orientada a Objetos**

1. **Defina lo que es un objeto.**

Se compone por atributos y métodos.

1. **Suponiendo que tenemos el objeto “Ratón” ¿cuáles son sus propiedades y operaciones disponibles específicas que lo describen?**

Atributos podrían ser número de botones, tipo de conexión (alámbrica o inalámbrica), material, cantidad de piezas a utilizar, marcas.

Métodos podría ser para que dispositivos puede servir, en que superficies funcionan mejor, para que sirve o para que esta programado cada botón, como se limpia.

1. **Defina abstracción.**

Es cuando un método no puede ser definido en una clase en específica ya que no está lo suficientemente conceptualizado como para poder definir el método.

Ejemplo: un método llamado alimentarse, si lo analizamos con un objeto persona y un objeto animal, ambos tienen diferentes maneras de alimentarse como que el humano cocina el alimento y el animal tiene que cazar para comer.

1. **Defina lo que es una clase en Programación.**

Una clase es como se define a un objeto, dentro de la clase de declaran los atributos, métodos y parámetros que puede ocupar el objeto para ser creado al inicializar el programa.

1. **¿Qué es un constructor de clase?**

Inicializa los atributos de la clase, a variables que el usuario indique o automáticamente lo compara a variables nulas (0). Puede existir la sobrecarga de constructores donde existen más de un constructor paro se diferencian por los atributos a declarase.

1. **Mencione diferencias entre Programación Estructurada y Programación Orientada a Objetos.**

La programación estructurada, como el nombre lo dice, es una programación que va línea por línea de ejecución hasta finalizar el programa. El poo es una manera más normal de entender o de programar, ya que se programa un objeto como la persona lo entiende y busca atributos del objeto para poder ser creado.

1. **Defina encapsulamiento y menciones cuáles son sus ventajas.**

El encapsulamiento es cuando es private la variable lo que nos indica que de otra clase o de otro paquete no se pueden alterar dichas variables que tiene el objeto, si el tipo es público, a esa variable pública se puede acceder desde cualquier otra clase o método de otros paquetes, pudiendo alterar el contenido de la misma. Lo cual en forma private, evita esta alteración.

1. **¿Qué es la sobrecarga de métodos? De un ejemplo.**

La sobrecarga de métodos es cuando existen dos métodos llamados de la misma forma la diferencia entre dichos métodos son los parámetros con los que se declaran, ejemplo de uns sobrecarga del método salidar

Public void saludar() //sin parametors

{

System(“Hola”);

}

Public void saludar(string nombre) //con parametors

{

System(“Hola mi nombre es: ”+nombre);

}

1. **¿Qué es la sobre escritura de métodos? De un ejemplo**

Cuando el mismo método se sobrescribe en otra clase cambiando el contenido o los parámetros.

1. **Practica en programación: Utilizando un proyecto llamado Examen**

* **Crear una clase “Caja” que contenga los siguientes atributos: Ancho, Alto, Profundidad a los cuales solo se puede tener accesos dentro de la misma clase.**
* **Se pueda crear un objeto usando 2 constructores: El primero tiene que recibir como parámetro 3 valores para posteriormente asignarlos a los atributos globales anteriores y un segundo constructor por default.**
* **Implementar encapsulamiento para los todos los atributos**
* **Crear método volumen() el cual calcule el volumen de la caja y almacene el resultado en otro atributo.**
* **Sobre escribir el método toString() que retorne la representación del objeto (la representación en cadena tiene que contener ancho, alto, profundidad y volumen).**
* **Sobre escribir el método equals() el cual reciba como parámetro un objeto Object y sea capaz de discernir si el objeto recibido es equivalente al objeto desde donde se ejecuta el método equals(). (Este método retorna un boolean)**
* **Crear una clase principal para crear 2 objetos tipo Caja utilizando los diferentes constructores. Una vez creados los objetos imprimir su representación en cadena. Para finalizar utilizando el método equals() imprimir si los objetos son iguales.**
* **Comentar código (atributos, métodos ,etc)**

**Suerte!**