

1. ¿Por qué la programación estructura recibe ese nombre?

La programación estructurada obtuvo su nombre porque se basa en Construcciones de flujo de control como bucles y condicionales organizar y controlar el flujo de ejecución del programa, no Permite saltos incondicionales como algunos lenguajes de programación.
programación más antigua.

2. ¿Qué es Programación Orientada a Objetos (POO)?

La Programación Orientada a Objetos (POO) es un paradigma programación donde residen los datos y el comportamiento Encapsulados juntos en objetos, los programas están organizados Términos de interacción entre estos objetos.

3. ¿Qué es la herencia en la POO?

La herencia en OOP se refiere a la capacidad de una clase para heredar Otras propiedades y métodos de clase. Esto le permite crear nuevas clases. Aquí hay una variante de una clase existente para que sea más fácil Reutilización de código y organización de programas.

4. ¿Cuáles son los pilares de la POO?

Qué los pilares de la programación orientada a objetos son la encapsulación, la herencia y el polimorfismo. La encapsulación se refiere a la capacidad de un objeto para ocultarse Solo una interfaz pública es visible para el modo interno. heredar Como se mencionó en la pregunta anterior, el polimorfismo se refiere a La capacidad de responder de manera diferente a diferentes clases de objetos. diferente del mismo mensaje

5. ¿Qué es un puntero?

Un puntero es una variable que contiene una dirección de memoria. Otra variable. Los punteros son útiles para manipular estructuras de cadenas.
Datos dinámicos y paso de parámetros por referencia

6. ¿Cuál es el Operador de dirección en un Puntero?

1. ☒ a) & b) * c) == d) Ninguno

7. ¿Cuál es el Operador de Indirección?

1. & ☒ b) * c) == d) Ninguno

8. Las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa. ¿Cuáles son?

- Estructuras de control de bucles: for, while do while
- Estructuras de control de saltos: break, goto, continue;
- Estructuras de control condicionales: if-else, switch