



1. Si tienes una clase llamada 'vehículo', ¿qué tipo de datos podrías incluir como atributos?
  - A) Aceleración y freno.
  - B) Ruedas y motor.
  - C) Color, marca y velocidad.
  - D) Conductor y pasajeros.
  
2. ¿Cómo se relacionan las instancias de objetos creados a partir de la misma clase?
  - A) No pueden tener métodos diferentes.
  - B) No pueden ser creadas al mismo tiempo.
  - C) Comparten la misma lógica pero tienen estados independientes.
  - D) Son idénticas en todos los aspectos.
  
3. ¿Cuál es un beneficio de usar la Programación Orientada a Objetos?
  - A) Complica la estructura del código.
  - B) Aumenta la duplicación de código.
  - C) Facilita la corrección de errores en el código.
  - D) Elimina la necesidad de clases.
  
4. ¿Qué significa encapsulación en la Programación Orientada a Objetos?
  - A) Es la eliminación de atributos de una clase.
  - B) Es el acceso a datos a través de métodos privados.
  - C) Es la creación de múltiples clases.
  - D) Es la creación de instancias de un objeto.
  
5. ¿Cuál fue uno de los primeros lenguajes en implementar la Programación Orientada a Objetos?
  - A) C++.
  - B) Python.
  - C) Simula 67.
  - D) Java.
  
6. ¿Cómo ayuda la encapsulación en la programación orientada a objetos?
  - A) Elimina la necesidad de herencia.
  - B) Facilita la duplicación de código.
  - C) Oculta información innecesaria para el usuario.
  - D) Permite que todos los datos sean públicos.
  
7. ¿Cuál es un beneficio del polimorfismo en la programación orientada a objetos?
  - A) Complica la estructura del programa.
  - B) Permite que un objeto adopte múltiples formas.
  - C) Evita la reutilización de código.



- D) Reduce la legibilidad del código.
8. ¿Por qué es importante la fiabilidad en la programación orientada a objetos?
- A) Facilita el análisis de cada módulo.
  - B) Elimina la necesidad de documentación.
  - C) Permite la creación de programas más complejos.
  - D) Reduce la necesidad de pruebas.
9. ¿Qué desventaja se menciona sobre la programación orientada a objetos?
- A) Necesita una extensa documentación.
  - B) No permite la creación de módulos.
  - C) Es más fácil de entender para los nuevos programadores.
  - D) Requiere menos tiempo de desarrollo.
10. ¿Cómo afecta la programación orientada a objetos al desarrollo de software?
- A) Facilita el desarrollo y la comprensión del código.
  - B) Elimina la necesidad de pruebas.
  - C) Lo hace más complicado y menos eficiente.
  - D) Aumenta el tiempo de desarrollo.
11. ¿Por qué es importante la legibilidad del código en Python?
- A) Reduce la necesidad de comentarios.
  - B) Aumenta la complejidad del código.
  - C) Limita el uso de bibliotecas externas.
  - D) Facilita la colaboración entre programadores.
12. ¿Qué paradigma de programación NO es comúnmente asociado con Python?
- A) Programación orientada a objetos.
  - B) Programación funcional.
  - C) Programación imperativa.
  - D) Programación de ensamblador.
13. ¿Qué permite hacer la plataforma OnlineGDB?
- A) Instalar software en la computadora.
  - B) Escribir, compilar y ejecutar código en varios lenguajes.
  - C) Crear aplicaciones de escritorio sin conexión a internet.
  - D) Programar solo en Python.
14. ¿Qué se necesita para utilizar OnlineGDB?
- A) Un software específico instalado.



- B) Un navegador web y conexión a internet.
- C) Un conocimiento avanzado de programación.
- D) Un dispositivo móvil exclusivamente.

15. ¿Cuál es una ventaja de usar OnlineGDB en comparación con la programación tradicional?

- A) Solo se puede usar en un solo lenguaje.
- B) Requiere instalación de software.
- C) No permite colaboración entre programadores.
- D) Es accesible desde cualquier lugar con internet.