|  |
| --- |
| 1. 下列哪一個不是Java 語言的特性:   (A) 繼承 (B) 多重繼承 (C)多型 (D)封裝 |
| 1. Java 八大基本型別不包含那一種:   (A) String (B) char (C) boolean (D)long |
| UML 圖如下: |
| 1. 請問Employee 中的name 修飾子應宣告為何種   (A) private(B) protected (C) public (D)不用加修飾子 |
| 1. 請問Employee 與Teacher的關係為何   (A) 繼承 (B) 實作 (C)多型 (D)封裝 |
| 1. 今天如果要使用Teacher 建構物件teacher該如何宣告    1. Teacher teacher ;    2. Teacher teacher =new Teacher()    3. Teacher teacher = new Teacher(“David”,”15000”,”1200”,”1.5”,”3000”);    4. Teacher teacher = new Teacher(“David”,15000,1200,1.5,3000) |
| 1. 下面介面程式片段中 有幾個語法錯誤   interface Action {  public void do\_something(){}; }  public class Teacher extends Action{  public static void main(String[] args) {  } }   1. 沒有錯誤 (B)1個 (C) 2個 (D) 3個 |
| 1. 請試著修正上述語法錯誤並寫出正確的程式碼 2. 參考下面程式片段，撰寫一個靜態方法MathToolKit ，其中包含一個計算平均值的方法。並撰寫一個MathTool 的類別，在Main 方法中呼叫。   class MathToolKit{  public \_(A)\_\_ \_(B)\_\_ average(\_(C)\_){    } }  public class MathTool {  public static void main(String[] args){  int a[] = new int[100];  double avg = MathToolKit.average(a)  } }  請問\_(A) 、(B)、(C)處應該填寫什麼?  A\_\_\_\_\_\_\_\_\_\_\_  B\_\_\_\_\_\_\_\_\_\_\_\_  C\_\_\_\_\_\_\_\_\_\_\_\_ |
| 1. 9. 參考下面抽象類別程式片段，有幾個語法錯誤   abstract class Car {  public void drive() ; } public class Toyota implements Car {  @Override  public void drive() {  System.*out*.println("I drive 100 km/hour");;  } }  (A)沒有錯誤 (B)1個 (C) 2個 (D) 4個   1. 10.承上題: 請試著修正上述語法錯誤並寫出正確的程式碼 |
| 11-18 為題組，會使用到下面的類別  class Car {  private int tires;  public Car(int tires){  this.tires = tires;  }   public int getTires(){return this.tires;}  public void drive(){  System.*out*.println("I drive 90km/hours");  }  } |
| 1. 參考下面程式片段，請問輸出為何   public class Toyota extends Car {  public static void main(String[] args) {  Car altis = new Toyota();  System.*out*.println(altis.tires);  } }   * 1. 編譯時期異常 (B) 0 (C) 4 (D) 以上皆非 |
| 1. 參考下面程式片段，請問輸出為何   public class Toyota extends Car {   public Toyota(int tires){  super(4);  }   public static void main(String[] args) {  Car altis = new Toyota(0);  System.*out*.println(altis.getTires());   } }  (A)編譯時期異常 (B) 0 (C) 4 (D) 以上皆非 |
| 1. 參考下面程式片段，請問輸出為何   public class Toyota extends Car {   public Toyota(){  /\*\*/  }   public Toyota(int tires)  {  super(tires);  System.*out*.println(super.tires);  }   public static void main(String[] args) {  Car altis = new Toyota();  }  }  Toyota 的建構子中要如何填寫，讓程式可以正常完成  (A).super() .tire  (B) this.tire=4;  (C) this(4);  (D) 以上皆可 |
| 1. 參考下面程式片段，如要使程式正常執行，該如何宣告   public class Toyota extends Car {   public Toyota(int tires)  {  super(tires);  System.*out*.println(super.tires);  }   public void turbo(){  System.*out*.println("I drive 200 km/hour");  }   public static void main(String[] args) {  /\* 宣告 \*/  altis.turbo();  }  }   * 1. Car altis = new Toyata(4);   2. Toyota altis = new Car(4);   3. Car altis = new Car(4);   4. Toyota altis = new Car(); |
| 1. 參考下面程式，請問可能輸出為何   public class Toyota extends Car {   public Toyota(int tires)  {  super(tires);  }   public void turbo(){  System.*out*.println("I drive 200 km/hour");  }   public static void main(String[] args) {  Car altis = new Toyota(4);  altis.drive();  }  }   * 1. 沒有輸出   2. 編譯時期異常   3. I drive 200 km/hour   4. I drive 90km/hours |
| 1. 參考下面程式，請問可能輸出為何   public class Toyota extends Car {   public Toyota(int tires)  {  super(tires);  }    public void drive(){  System.*out*.println("I drive 200 km/hour");  }   public static void main(String[] args) {  Car altis = new Toyota(4);  altis.drive();  }  }   * 1. 編譯時期異常   2. I drive 90km/hours   3. I drive 200 km/hour   4. I drive 90km/hours   I drive 200km/hours |
| 1. 參考下面程式，請問可能輸出為何   public class Toyota extends Car {   public Toyota(int tires)  {  super(tires);  }   public void drive(){  System.*out*.println("I drive 200 km/hour");  }   public static void main(String[] args) {  Car altis = new Car(4);  altis.drive();  }  }   * 1. 編譯時期異常   2. I drive 90km/hours   3. I drive 200 km/hour   4. I drive 90km/hours   I drive 200km/hours |
| 1. 參考下面程式，請問可能輸出為何   public class Toyota extends Car {   public Toyota(int tires)  {  super(tires);  }    public void drive(){  System.*out*.println("I drive 200 km/hour");  }   public static void main(String[] args) {  Toyota altis = new Car(4);  altis.drive();  }  }   * 1. 編譯時期異常   2. I drive 90km/hours   3. I drive 200 km/hour   4. I drive 90km/hours   I drive 200km/hours |

參考解答

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| B | A | B | A | D |
| 6 | 7 | 8 | 9 | 10 |
| D | 略 | A:static  B double  C:int [] a | C | 略 |
| 11 | 12 | 13 | 14 | 15 |
| A | C | C | A | D |
| 16 | 17 | 18 |  |  |
| C | B | A |  |  |

註: 第6題

1,介面不能有body

2.介面不試用extends，要implements

3. do\_something() 需要被實作

第9題

1. drive() 方法沒有body 需宣告為抽象
2. Toyota 需要用extends 而非implements