**QUIZ**



|  |  |  |  |
| --- | --- | --- | --- |
| 1. Use the diagram above to determine if the following declarations are valid or invalid | | | |
| a. | PC a = new PC(); | Valid |  |
| Invalid |  |
| b. | Game b = new PC(); | Valid |  |
| Invalid |  |
| c. | Card c = new Tabletop(); | Valid |  |
| Invalid |  |
| d. | Object d = new Card(); | Valid |  |
| Invalid |  |
| e. | Card e = d; | Valid |  |
| Invalid |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Use the diagram above and the declarations provided to determine if the following statements will print out true or false:  * Game g = new Game(); * Console c = new Console(); * Game z = new Console(); | | | |
| a. | System.out.println(g instanceof Game); | True |  |
| False |  |
| b. | System.out.println(g instanceof Console); | True |  |
| False |  |
| c. | System.out.println(c instanceof Console); | True |  |
| False |  |
| d. | System.out.println(c instanceof Game); | True |  |
| False |  |
| e. | System.out.println(z instanceof Game); | True |  |
| False |  |
| f. | System.out.println(z instance of Console); | True |  |
| False |  |

|  |  |  |
| --- | --- | --- |
| 1. If the method ‘greeting()’ is defined for both the ‘Birthday’ and ‘AdultBirthday’ classes, which method will run for the given code:  * Birthday happy;   happy = new AdultBirthday(“Joe”, 39);  happy.greeting(); | | |
| a. | The one defined for ‘*Birthday’* because that is the type of the variable ‘*happy’*. |  |
| b. | The one defined for ‘*AdultBirthday*‘ because that is the type of the object referred to by ‘*happy’*. |  |
| c. | The one closest in the source code to the happy.greeting() statement. |  |

|  |  |  |
| --- | --- | --- |
| 1. The class ‘Rodent’ has a child class ‘Rat’ and another child class ‘Mouse’ and the class ‘Mouse’ has a child class ‘PocketMouse’. Given the declarations below, which of the following will cause a compiler error:  * Rodent rod; * Rat rat = new Rat(); * Mouse mos = new Mouse(); * PocketMouse pkt = new PocketMouse(); | | |
| a. | rod = rat; |  |
| b. | rod = mos; |  |
| c. | pkt = null; |  |
| d. | pkt = rat; |  |

|  |  |  |
| --- | --- | --- |
| 1. Using the same situation as above, which of the following array declarations is correct for an array that is expected to hold up to 10 objects of types ‘Rat’, ‘Mouse’, and ‘PocketMouse’ | | |
| a. | Rat[] array = new Rat[10]; |  |
| b. | Rodent[] array = new Rat[10]; |  |
| c. | Rodent[] array = new Rodent[10]; |  |
| d. | Rodent[10] array; |  |

|  |  |  |
| --- | --- | --- |
| 1. Using the code below, chose the correct statement:  * class GameShape {   public void displayShape() {  System.out.println("displaying shape");  }  }   class PlayerPiece extends GameShape {  public void movePiece() {  System.out.println("moving game piece");  }  }   public class TestShapes {  public static void main (String[] args) {  GameShape shape = new PlayerPiece(); shape.displayShape(); shape.movePiece();  }  } | | |
| a. | PlayerPiece class inherits the displayShape() method |  |
| b. | GameShape class inherits the displayShape() method |  |
| c. | GameShape class inherits the movePiece() method |  |
| d. | PlayerPiece class inherits the movePiece() method |  |

|  |  |  |
| --- | --- | --- |
| 1. Using the code below, chose the correct output:  * class Hotel{   public int bookings = 2;  public void book(){  bookings++;  }  }  public class SuperHotel extends Hotel{  public void book(){  bookings--;  }  public void book(int size){  book();  super.book();  bookings += size;  }  public static void main(String args[]){  SuperHotel s = new SuperHotel();  s.book(2);  System.out.print(s.bookings);  }  } | | |
| a. | Compile error |  |
| b. | 2 |  |
| c. | 4 |  |
| d. | 3 |  |