Diem Pham

Dtp160130

|  |
| --- |
| 1. Examine the following source code.  Assume this is a complete file named Epona.java.  Which of the items below would be treated as a syntax error when compiling the code?  Select all that apply.  1.5 out of 3 points  import java.util.\*;  package Hyrule;    public class Epona  {  void setGallopSpeed(int x)  {   gallopSpeed = x;   }  static private int gallopSpeed;  private double weight;  public void speedUp()  {   this.speedUp(10);   }  public void speedUp(int x)  {  this.setGallopSpeed(x);   }  } |
| |  | | --- | |  | |  | |  | |  | |  | | Answers: | use of this to call a non-overloaded function | |  | placement of class variables | |  | lack of access specifiers for function | |  | use of this to call overloaded function | |  | **package statement** | |  | use of static for variables | |  | importing an unused library | |

|  |
| --- |
| 4. Which of the things below are **TRUE** of the abstract keyword.  Select all that apply.  3 out of 4 points |
| |  | | --- | |  | |  | |  | | Answers: | prevents a subclass from overriding the abstract method | |  | **prevents a class from being instantiated** | |  | prevents the class from being used as a data type | |  | prevents a class from being extended | |  | prevents a class from defining constructors | |  | **prevents a method from being defined in the class**  Explain: 13.2.2 in chapter 13 page 500 | |
| 8. Which of the following statements are **TRUE** of immutable objects?  Select all that apply.  5 out of 6 points |
| |  |  | | --- | --- | | Answers: | All class methods must be static | |  | **Accessor methods cannot return a class variable reference to an object that can be modified** | |  | The class variables must be of a primitive data type | |  | **There cannot be any mutator methods for the class variables** | |  | **All class variables must be private** | |  | The class must be declared as abstract  Explain: 9.12 in chapter 9 page 353 | |
| 10. Which of the following statements about recursion are **TRUE**? Select all that apply.  2 out of 6 points |
| |  |  | | --- | --- | | Answers: | **Every solution that can be solved by using loops can also be solved recursively** | |  | Recursion only happens when a function calls itself | |  | There are no drawbacks to using recursion. | |  | **Recursion involves the repeated execution of a segment of code.** | |  | Recursive solutions are always simpler than iterative solutions | |  | **Infinite recursion could happen if there were an infinite amount of memory available**  Explain: 18.9 in chapter 18 page 726 | |
| 12. Which of the following function headers is syntactically correct?  0 out of 3 points |
| |  |  | | --- | --- | | Answers: | public static void HeyListen (int... fairies, String... Navi) | |  | public static void HeyListen (int... fairies, String Navi) | |  | **public static void HeyListen (String Navi, int... fairies)** | |  | public static int... HeyListen (int fairies, String Navi) | |

Explain: chapter 6 page 204

|  |
| --- |
| 14. Given the following code, identify which variables are accessible inside of the Knight class without using acessors or mutators.  Select all that apply.  4.5 out of 6 points  package Hyrule  public class SkullKid  {  String name;  protected int age;  //constructors, mutators, accessors  }    package Hyrule  public class Stalfos  {  SkullKid SK = new SkullKid  protected String weapon;  public int boneCount;  //constructors, mutators, accessors  }    package Enemy  public class Knight extends Stalfos  {  boolean shield;  private boolean helmet;  //constructors, mutators, accessors  } |
| |  | | --- | |  | |  | |  | | Answers: | name | |  | age | |  | **weapon** | |  | **boneCount** | |  | **shield** | |  | **helmet** |   Explain: 6.9 in chapter 6 page 222 |
| 15. Which of the following statements are **FALSE**?  Select all that apply.  1 out of 4 points |
| |  | | --- | |  | |  | | Answers: | When space for an array is allocated, each element is assigned a default value. | |  | **Declaring an array variable allocates space for the array in memory** | |  | Non-constant values can be used to declare the size of an array. | |  | **When used with an array variable, the assignment operator (array1 = array2) creates a deep copy.** | |  | An array can be created without explicitly stating the size of the array | |  | Arrays can be returned from a function  Explain: 6.9 chapter 6 page 222 | |
| 16. Which of the following statements about interfaces are **FALSE**? Select all that apply.  0.5 out of 4 points |
| |  |  | | --- | --- | | Answers: | **the interface can be instantiated by using the new keyword** | |  | all methods are public | |  | **all variables in the interface are abstract** | |  | access specifiers are optional when declaring the interface members | |  | all methods are abstract | |  | **only one interface can be implemented per class** | |  | **all methods are static** | |
| Explain: 13.5 chapter 13 page 506  17. What is the output of the following code segment?  0 out of 3 points    public static void rupees(int blue, int red, int green)  {  blue += 5;  red = 20;  green -= green;  }  public static void main()  {  int r=0, b=1, g=2;  rupees(b, r, g);  System.out.println(r + b + g);  } |
| |  |  | | --- | --- | | Answers: | **3** | |  | 0 | |  | 25 | |  | 26 | |
| Explain: print out sum of r, b and g  19. Which of the following things are accomplished by the final keyword?  5 out of 6 points |
| |  |  | | --- | --- | | Answers: | **prevents a class from being extended** | |  | initiates the last iteration of a loop | |  | **prevents a method from being overridden** | |  | prevents a method from being overloaded | |  | prevents an interface from being implemented | |  | **prevents a variable from being modified** | |
| Explain: 9.7 chapter 9 page 377  20. Which of the following code segments will produce a syntax error? Select all that apply.  2.25 out of 3 points |
| |  | | --- | |  | |  | | Answers: | int hearts = 3;  hearts += 2.5; | |  | **float potion = 3.5;**  **int hearts = (int) potion;** | |  | char letter = 'L';  System.out.println((char)((int)letter)); | |  | **long sword = 10;**  **int arrow = sword;** | |  | char letter = 'Z';  if (letter >= 'A' && letter <= 'Z')  System.out.println(letter + "elda"); | |  | String rupees = '1' + "00"; | |

Explain: can’t assign long to int

3.5 should be 3.5f