|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | Describe the purpose of a Java class constructor. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | The java constructor's purpose defines the actions that occur upon creating the object. If a constructor is not defined that the object will be initialized with null values in the fields.   They are really nothing more than the initial methods that occur on every object from the moment they are instantiated. | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | What is the purpose of set<attribute> and get<attribute> methods? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | A setter or set<attribute> is used to set private fields within an instantiated class object, this ensures that one object cannot modify another object's fields unintended.   A getter or get<attribute> returns the value of a field within an object without modifying it in anyway. If the value is an integer it return and integer and so on. | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | Provide a definition of the term ***semantics.*** | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Semantics are the rules and definitions that form the syntax of a specific language and the interaction of the primitive values that are invoked within the program.   In other words semantics are the rules that govern all computation that occurs mathematically and syntactically within the specific programming language. | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | Explain the term ***instance variable.*** | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | An instance variable is a variable defined within a class but outside a method. These are fields are really the public fields that are instantiated within another class and are invoked or initialized outside the class. Instance variables are not private fields, and private fields are not instance variables. | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | Given the initialization of str as:  String str = "Arizona State University";  give the value for each of the following expressions:      str.charAt (8)      str.indexOf ("i")      str.substring (7)      str.substring (3, 5)      str.lastIndexOf (“i”) | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | S 2  State University zo 21 | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | Provide the output for the following program:  public class StringMutation {      public static void main (String [] args) {          String phrase = “Pirates of the Caribbean”;          String mutation1, mutation2, mutation3, mutation4;          System.out.println (“Original string: \”” + phrase + “\””);          System.out.println (“Length of string: “ + phrase.length ());          mutation1 = phrase.concat (“: Curse of the Black Pearl”);          mutation2 = mutation1.toUpperCase ();          mutation3 = mutation1.replace ('r', 'X');          mutation4 = mutation3.substring (3);          System.out.println (“Mutation #1: “ + mutation1);          System.out.println (“Mutation #2: “ + mutation2);          System.out.println (“Mutation #3: “ + mutation3);          System.out.println (“Mutation #4: “ + mutation4);      }  } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Original string: "Pirates of the Caribbean" Length of string: 24 Mutation #1: Pirates of the Caribbean: Curse of the Black Pearl Mutation #2: PIRATES OF THE CARIBBEAN: CURSE OF THE BLACK PEARL Mutation #3: PiXates of the CaXibbean: CuXse of the Black PeaXl Mutation #4: ates of the CaXibbean: CuXse of the Black PeaXl | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **5 of 5** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | Assuming that we wish to represent an mp3 player as a software object: a. List 5 characteristics that represent its state (attributes): b. List 5 characteristics that represent its behavior (methods): | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | public class MP3 { //List 5 characteristics that represent its state (attributes): private String myName; private boolean powerStatus; private String myCurrentSong; private int numberOfSongsTotal; private int remainingFreeSpace; private List<List<String>> musicLibrary = new ArrayList<List<String>>(); //List 5 characteristics that represent its state (attributes): public MP3(){ this.myName = ""; this.powerStatus = false; this.myCurrentSong = ""; this.numberOfSongsTotal = 0; this.remainingFreeSpace = 32; } void powerOn(){ this.powerStatus = true; } void setMyName(String input){ this.myName = input; } void setMyCurrentSong(String input){ this.myCurrentSong = input; } void setNumberOfSongsTotal(int input){ this.numberOfSongsTotal = input; } void setRemainingFreeSpace(int input){ this.remainingFreeSpace = input; } void loadSongOntoMP3(String name, int spacetaken){ this.musicLibrary.add(name); this.remainingFreeSpace = this.remainingFreeSpace - spacetaken; } | |  | Instructor Explanation: | 2 points each question | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **4 of 4** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | Implement (define) a class named StockType.java. A StockType has a symbol (a String), a type (an int) and a price (a double). It provides the methods to set the price (give the price an initial value), change the price (increase or decrease the price by some value), to return the current price, to set the symbol value, to return the symbol value, to set the type value and return the type value. Objects that are declared to be of StockType are given an initial price of 0, an initial type of 0 and an initial symbol value of the empty string. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | public class StockType { private String symbol; private int type; private double price; public StockType(){ this.symbol = ""; this.type = 0; this.price = 0; } void setSymbol(String input){ this.symbol = input; } void setType(int input){ this.type = input; } void addToPrice(double input){ this.price = this.price + input; } void subtractFromPrice(double input){ this.price = this.price - input; } String getSymbol(){ return this.symbol; } int getType(){ return this.type; } double getPrice(){ return this.price; }     public static void main (String [] args) {          } } | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **5 of 5** | |  | Comments: |  | | |