The Walker School - Technology Department – AP Computer Science

Pre-Barons Review Questions for Classes

Concepts Covered: Class signatures, instance fields, constructors, mutator methods, accessor methods, toString() methods

Class signatures

Concept to Know:

* Classes in Java are blueprints for objects in the real world.
* Classes are named after a major object.

1. Write a class signature for a Car class.

**public class Car{}**

1. Write a class signature for a Bank Account class.

**public class BankAccount{}**

1. Write a class signature for a Dress class.

**public class Dress{}**

1. Write a class for a Die class.

**public class Die{**

**private int faceValue; //holds the value of the die**

**public void setValue(int val){**

**faceValue = val;**

**}**

**public int getValue(){**

**return faceValue;**

**}**

**public void rollDie(){**

**faceValue = (int)(Math.random() \* 6) + 1;**

**}**

**public String toString(){**

**return “Die Value: ”+faceValue;**

**}**

**}**

1. Write a class for a Train class.

**public class Train{**

**String Destination, Origin;**

**public Train(String Destination, String Origin){**

**this.Destination = Destination;**

**this.Origin = Origin**

**}**

**public String getDestination(){return Destination;}**

**public String getOrigin(){return Origin;}**

**}**

Instance Fields

Concepts to Know:

* The attributes of an object are represented by the instance fields of a class.
* Instance fields can be a primitive data type (boolean, int, float, double, char) or an object, such as a String, list, or an ArrayList, or can be an instance of another class.

1. Write three attributes of a car. Each field should be private in scope.

**private int hoursepower;**

**private String brandName;**

**private boolean EngineOn;**

1. Write three attributes of a bank account. Each field should be private in scope.

**private double totalAmount;**

**private String accountHolderName;**

**private int accountNumber;**

1. Write three attributes of a dress. Each field should be private in scope.

**private double price;**

**private boolean onSale;**

**private int size;**

1. Write three attributes of a typical die. Each field should be private in scope.

**private int value, numOfSides;**

**private boolean isEvenNumber;**

1. Write three attributes of a train. Each field should be private in scope.

**private int milesTraveled, milesRemaining;**

**private String destination;**

Constructors

Concepts to Know:

* Constructors instantiate an instance of the class. They are the cookie-cutters of the objects of the classes they represent.
* Constructors should set each of the instance fields to their default value.
* You need to pass the instance fields into the constructor; they need to be different names than the actual instance fields for the class.
* You can use this.instancefieldname or change the name slightly, such as theColor.

1. Write a constructor for a Car class.

**public Car(String make, int bhp){**

**horsepower = bhp;**

**brandName = make;**

**EngineOn = false;**

**}**

1. Write a constructor for a Bank Account class.

**public BankAccount(String AccountHolder, int accountNumber){**

**AccountHolderName = AccountHolder;**

**this.accountNumber = accountNumber;**

**balance = 0;**

**}**

1. Write a constructor for a Dress class.

**public Dress(int dressPrice, int dressSize){**

**size = dressSize;**

**price = dressPrice;**

**onSale = false;**

**}**

1. Write a constructor for a Die class.

**public Die(int sides){**

**faceValue = 1;**

**isEven = false;**

**numOfSides = sides;**

**}**

1. Write a constructor for a Train class.

**public Train(String trainDestination){**

**destination = trainDestination;**

**milesTraveled = 0;**

**milesRemaining = 0;**

**}**

Accessor Methods

Concepts to Know:

* Assessor methods allow the client to get information from a class, such as the size of a dress in stock.

1. Write an assessor method for a Car class instance variable.

**public int getBHP(){**

**return horsepower;**

**}**

1. Write an assessor method for a Bank Account class instance variable.

**public int getBalance(){**

**return balance;}**

1. Write an assessor method for a Dress class instance variable.

**public boolean onSale(){**

**return onSale;}**

1. Write an assessor method for a Die class instance variable.

**public int getValue(){**

**return value;}**

1. Write an assessor method for a Train class instance variable.

**public String getDestination(){**

**return destination;}**

Mutator Methods

* Mutator methods allow the client to change the value of a variable in a class, such as total.

1. Write a mutator method or a Car class instance variable.

**public void setPrice(int newPrice){**

**price = newPrice;**

**}**

1. Write a mutator method for a Bank Account class instance variable.

**public void setValue(double newValue){**

**value = newValue;**

**}**

1. Write a mutator method for a Dress class instance variable.

**public void setSale(boolean sale){**

**onSale = sale;**

**}**

1. Write a mutator method for a Die class instance variable.

**public void setValue(int newValue){**

**value = newValue;**

**}**

1. Write a mutator method for a Train class instance variable.

**public void setMilesTraveled(int newMilesTraveled){**

**milesTraveled = newMilesTraveled ;**

**}**

toString Methods

Concepts to Know:

* toString methods return data about the class in a String format.

1. Write a toString() method or a Car class.

**public String toString(){**

**return “Price: ”+price+”\tBrand: ”+brandName+”\tHorse Power: ”+horsepower;**

**}**

1. Write a toString() method for the Bank Account class.

**public String toString(){**

**return accountNumber“\tBalance: ”+accountBalance;**

**}**

1. Write a toString() method for the Dress class

**public String toString(){**

**return itemNumber+“\tSale Status”+onSale+”\tPrice”+price;**

**}**

1. Write a toString() method for the Die class.

**public String toString(){**

**return “Dice Value: ”+value;**

**}**

1. Write a toString() method for a Train class.

**public String toString(){**

**return “Train Going To: ”+destination+”\tMiles Traveled: ”+distanceTraveled;**

**}**