ORACLE
Academy

academy.oracle.com

Java Fundamentals
7-1: Classes, Objects, and Methods
Practice Activities

Objectives:

- Recognize the correct general form of a class
- Create an object of a class
- Create methods that compile with no errors
- Return a value from a method
- Use parameters in a method
- Create a driver class and add instances of Object classes
- Add a constructor to a class
- Apply the new operator
- Describe garbage collection and finalizers
- Apply the this reference
- Add a constructor to initialize a value

Vocabulary:

Identify the vocabulary word for each definition below.

	A template used for making Java objects.
	An optional keyword used to access the members and methods of a class.
	An instance of a class.
	The operator used to create an instance of a class.
	A built-in function of the Java VM that frees memory as objects are no longer needed or referenced.
	A method that changes the state of an object.
	A method that returns information about an object back to the calling program.
	A procedure (changes the state of an object) or function (returns information about an object) that is encapsulated as part of a class.
	A verb used to describe the act of creating a class object using the keyword new.
	The process of assigning a default value to a variable.
	An object reference that has not been instantiated.
	An optional method that is called just before an object is removed by the garbage collector.
	The name of a variable that is associated with an object.
	A special method used to create an instance of a class.

Try It/Solve It:

1. Create a simple class Shape that will represent a 2-dimensional shape with line segments for edges. It should have the following instance variables: numSides (int), regular (boolean). Create at least two constructors and getter and setter methods.
2. Identify the key parts of the Java Class below. Put asterisks next to all the instance variables. Place a box around each constructor. Circle the signature of methods other than the constructor method. Place triangles around the parameters. Underline the return types of methods.

```
public class Animal  {
        int  weight,  height;
        double  speed;
        Animal() {
                weight = 50;
                height = 4;
                speed = 2;  //miles per hour
        }
        Animal(int w, int h, int s ) {
                weight = w;
                h = height;
                speed = s
        }
        public double getTime(double miles) {  //gets the number of hours to go these miles
                return miles/speed;
        }
        public int getWeight() {
                return weight;
        }
        public int getHeight() {
                return height;
        }
        public double getSpeed() {
                return speed;
        }
}
```

3. Write code to create two instances of the Animal class template listed in problem #2. Be sure to use each of the two constructors provided. Then add Java code that will print the following:

   a. Animal #1 has a speed of ___.

   b. Animal #2 has a speed of ___.

   Be sure that the blanks are automatically filled in with the actual speeds. Use the methods provided to access the speeds.

4. Write a class Student. It should have the following instance variables for the name, credits, grade point average (GPA), and quality Points. Create a constructor method. Create two other methods as follows:

   a. A method that will return the current grade point average which will be the quality points divided by the credits.

   b. A method that will take in the credits for a class or semester along with the quality points. It should update the credits, the quality points, and the GPA.

5. Using the class you created in #4, create three instances of the Student Class from the table below:

Name	**Credits**	**Quality Points**
Mary Jones	14	46
John Stiner	60	173
Ari Samala	31	69

6. Using the instance variables created in #5, add 13 credits and 52 quality points to the student "Ari Samala".

7. Using the Card class from the slides and test the program to make sure it works. Add a second random Card. Code is included below:

```
public class Card{
        String suit,name;
        int points;

        Card(int n1, int n2){
                suit = getSuit(n1);
                name = getName(n2);
                points = getPoints(name);
        }

        public String toString(){
        return "The " + name + " of " + suit;
        }

        public String getName(int i){
                if(i == 1)  return "Ace";
                if(i == 2)  return "Two";
                if(i == 3)  return "Three";
                if(i == 4)  return "Four";
                if(i == 5)  return "Five";
                if(i == 6)  return "Six";
                if(i == 7)  return "Seven";
                if(i == 8)  return "Eight";
                if(i == 9)  return "Nine";
                if(i == 10)  return "Ten";
                if(i == 11)  return "Jack";
                if(i == 12)  return "Queen";
                if(i == 13)  return "King";
                return "error";
        }

        public int getPoints(String n){
                if(n == "Jack" ||n == "Queen" ||n == "King"||n == "Ten")
                        return 10;
                if(n == "Two")
                        return 2;
                if(n == "Three")
                        return 3;
                if(n == "Four")
                        return 4;
                if(n == "Five")
                        return 5;
                if(n == "Six")
                        return 6;
                if(n == "Seven")
                        return 7;
                if(n == "Eight")
                        return 8;
                if(n == "Nine")
                        return 9;
                if(n == "Ace")
                        return 1;
                return -1;
        }

        public String getSuit(int i){
                if(i == 1)  return "Diamonds";
                if(i == 2)  return "Clubs";
                if(i == 3)  return "Spades";
                if(i == 4)  return "Hearts";
                return "error";
        }
}


public class Main {

        public static void main(String args[]){

                int suitNumber = (int)(Math.random()*4.0+1);
                int faceNumber = (int)(Math.random()*13.0+1);
                Card newCard = new Card(suitNumber,faceNumber);
                System.out.println(newCard);

        }

}
```

8. Add code to the Main class in exercise #7 to the following:
    a. Display the total point value for the two random cards.
    b. Ask the user if they would like another card. If they say yes display the new card and the points for all 3 cards in their "Hand".
    c. Loop to allow the user to continue to add cards to the hand until the number of points goes over 21 or the user decides not to add any more cards or the total number of cards is 5.