Montgomery College, CMSC 203
Worksheet 1
Module 7

Objectives

- Creating a Class
- Attributes , methods of the Class
- Data Element Class
- Driver Class

Concept Questions

1) A class specifies the ________ and ________ that a particular type of object has.
A) relationships; methods
B) fields; object names
C) fields; methods
D) relationships; object names
Answer: C

2) One or more objects may be created from a(n):
A) field
B) class
C) method
D) instance
Answer: B

3) Class objects normally have ________ that perform useful operations on their data(attributes).

A) fields
B) instances
C) methods
D) relationships
Answer: C

4) Which of the following are classes from the Java API?
A) `Scanner`
B) `Random`
C) `PrintWriter`
D) All of the above
Answer: D

5) Data hiding, which means that critical data stored inside the object is protected from code outside the object, is accomplished in Java by:
A) using the `public` access specifier on the class methods
B) using the `private` access specifier on the class methods
C) using the `private` access specifier on the class definition

D) using the `private` access specifier on the class fields
Answer: D

6) For the following code, which statement is NOT true?

```
public class Sphere
{
   private double radius;
   public double x;
   private double y;
   private double z;
}
```

A) `x` is available to code that is written outside the `Sphere` class.
B) `radius` is not available to code written outside the `Sphere` class.
C) `radius`, `x`, `y`, and `z` are called members of the `Sphere` class.
D) `z` is available to code that is written outside the `Sphere` class.
Answer: D

7) Methods that operate on an object's fields are called:
A) instance variables
B) instance methods
C) public methods
D) private methods
Answer: B

8) The scope of a `private` instance field is:
A) the instance methods of the same class
B) inside the class, but not inside any method
C) inside the parentheses of a method header
D) the method in which they are defined
Answer: A

9) It is common practice in object-oriented programming to make all of a class's:
A) methods private
B) fields private
C) fields public
D) fields and methods public
Answer: B
10) How can you find out if `DecimalFormat` is a Java API class or not?

Answer: `DecimalFormat` is a Java class and can be found in the list of classes in API documentation.

11) Write a `set` and `get` method for the `radius` attribute of the following `Sphere` class.

```
public class Sphere
{
    private double radius;
    public double x;
    private double y;
}
```

Answer:

```
     public void setRadius( double radius)
     {
         this.radius = radius;
     }
     public double  getRadius()

         return  radius;
     }
```

12) What happens if you don't create a constructor for a class?
A) Java won't compile the program
B) Java creates a default constructor
C) Java displays a warning
D) Nothing

Answer :B

13) Write a java statement based on the following class definition to create a new instance of the `Tree` class where the species is of type “elm”?

```
public class Tree {
     String species;
     public Tree (String species)
     {
          this.species = species;
     }
}
```

Answer :

```
Trees t1 = new Trees(“elm”);
```

14) Following code has compilation error. Why?

```
public class Cards {
     String suites;
     public Card (String suites)
     {
          this.suites = suites;
     }
}
```

Answer:
`Name mismatch between the class and the constructor`

15) Following is the declaration of the class Cards.

```
public class Cards {
     String suit;
     public void Cards(String suit)
     {
          this.suit = suit;
     }

     public String getSuit()
     {
          return suit;
     }
}
```

What is the output of the following code?

```
Cards c1 = new Cards ("Heart");
System.out.println(c1.getSuit());
```

Answer:
Compilation error when creating the instance.

Programming Questions

1) Write java code to define a class called Person (Your Data Element class) with the following information:

Fields

- `name (String)`
- `lastName(String)`
- `age (int)`

Methods of the `Person` Class

- `fullName`: this method does not take any parameter and it will return the full name of the a `Person's` object as a `String` value. For example if there is a `p1` Person object where the `name` field is `john` and the `lastName` field is `smith` then calling the method `fullName` for `p1` object should return `john smith`.

- `ageAfterTenYears`: this method does not take any parameter and it returns the age of a `Person's` object after 10 years. For example if there is a `p1 Person` object where the `age` field is 20 then calling the method `ageAfterTenYears` for `p1` object should return `30`.

- Create setter and getter methods for all the fields of the `Person` Class.

Test your `Persons` class with the following driver class and `WRITECODE` on the highlighted area shown below:

```
public class PersonDriver {

      public static void main(String[] args) {

            //Create a person object
            Person emily = new Person();
            //set name
            emily.setName("Emily");
            //set name
            emily.setLastName("Smith");
            emily.setAge(25);
            //Display the name for emily
            System.out.println( emily.getName());
            //Write code to display the full name for emily
            System.out.println( emily.fullName());
            //Write code to display the the age after ten years
            System.out.println( emily.ageAfterTenYears());

            /*WRITE CODE:
              create another object called bob with name "Bobby" and age 34.
              Call the similar methods for bob object and observe the results.
              What will be displayed for the full name of bob object? Why?
            */
            Person bob = new Person();
            bob.setName("Boby");
            bob.setAge(34);
            System.out.println( bob.getName());
            System.out.println( bob.fullName());
            System.out.println( bob.ageAfterTenYears());

      }

}

public class Person {
      private String name;
      private String lastName;
      private int age;


      public String fullName() {
            //String fullName = name + " " + lastName;
            return name + " " + lastName;
      }

      public int ageAfterTenYears() {
            return age+10;
      }
```

```
	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public int getAge() {
		return age;
	}

	public void setAge(int age) {
		this.age = age;
	}


}
```