**Topic 4: Classes**

Top of Form

Bottom of Form

**Classes in Java**

Recall classes from Week 9! Now we can create classes in Java.

The example below should look familiar. A class has fields and methods. In this case the fields are the String variables carrier and number. There is one special method called a **constructor**, we can identify in because it has the same name as the class. The constructor is called when a new object is created, the constructors job is to initialize the object.

|  |
| --- |
| public class Phone {       private String carrier;       private String number; // String for (515)-555-1234 format        public Phone(String inputCarrier, String inputNumber) {         carrier = inputCarrier;          number = inputNumber;       }        public String getCarrier() {         return carrier;       }       public void setCarrier(String carrier) {          this.carrier = carrier;       }       public String getNumber() {         return number;       }       public void setNumber(String number) {         this.number = number;      } } |

Now we are ready to create objects from the class, which we will do in main. We create an object using the keyword new followed by a call to the constructor. We can then use the object by calling any public non-static method.

|  |
| --- |
| public class Phone {       private String carrier;       private String number; // String for (515)-555-1234 format        public Phone(String inputCarrier, String inputNumber) {         carrier = inputCarrier;          number = inputNumber;       }        public String getCarrier() {         return carrier;       }       public void setCarrier(String carrier) {          this.carrier = carrier;       }       public String getNumber() {         return number;       }       public void setNumber(String number) {         this.number = number;      }      public static void main(String args[]) {         **Phone bobsPhone = new Phone("Verizon", "(515)-555-1234");**         System.out.println("Bob's phone number is ", **bobsPhone.getNumber()**);     } } |

[**Classes in Programming**](https://dmacc.blackboard.com/webapps/assignment/uploadAssignment?content_id=_3812345_1&course_id=_69150_1&group_id=&mode=view)

In the Week 10 Topic 2 assignment we created a class called Product that holds a stock number and price. Now create the Product class in Java.

* 1. The class should have a constructor that initializes stockNumber and price.
  2. The class should have getters and setters for stockNumber and price.
  3. In the main method create a new object using the constructor for Product.
  4. Use the getters from the new object to print information about the product.

Submit your .java file.

This is worth 20 points.

**Inheritance in Java**

Recall the concept of inheritance from Week 10, a class can be a subclass of another class. The subclass has the same fields and methods as the parent class.

In Java we use the keyword **extends** to make subclasses.

A screenshot of a social media post

Description automatically generated

[**Subclasses in programming**](https://dmacc.blackboard.com/webapps/assignment/uploadAssignment?content_id=_3812352_1&course_id=_69150_1&group_id=&mode=view)

Now create the Book class that inherits from Product. Don't forget that each class must go in a file with a matching name, so put the Book class in Book.java. In the main method create a new Book object. Use setters to set the author, title and numberPages. Use the getters to print information about the book.

A close up of a logo

Description automatically generated

Submit your .java file.

This is worth 10 points.