CS180, end of the course practice problems

Fill in the blanks

• A ________________ is called when an object is created, to initialize the object’s instance variables.

• true and false are constants of ________________ type

• ________________ methods receive the calling object as an implicit parameter.

• A ________________ is a template for an object.

• The ________________ statement terminates a method.

• The term overloading means

________________________________________________________________________
Methods

What is displayed when the following code segment is executed? Someone has created this very poorly written undocumented program. Do not try to deduce the purpose of this code. Simply show the output in the box on the bottom of the page. Show the values of participating variables, parameters and expressions for partial credit.

```java
public class Example {
    public static void main(String[] args) {
        int hour = 14;
        int m = 28;
        System.out.println("hour = " + hour + " minute = " + m);
        hour = myMethod(hour, m);
        System.out.println("hour = " + hour + " minute = " + m);
    }
    public static int myMethod(int h, int m) {
        if (m < 30 && h > 15 || m > 25) {
            h = h/3;
            m = m%2;
        } else {
            h = (m+1)/h;
            m = m % h;
        }
        System.out.println("h is " + h + " m is " + m);
        return h;
    }
}
```

```plaintext
hour = 14 minute = 28
hour = 7 minute = 28
```
Problem 2.

1. For the first part of the problem, develop a static method `computeChecksum()` that is going to be passed a parameter of type `String`. The method should verify that the string consists of digits and dashes only, and in such case it must return the sum of all digits in it.

   In case the parameter string contains any character other than a digit or a dash, the method should return -1.

Here are a few examples:

<table>
<thead>
<tr>
<th>Parameter string</th>
<th>Method should return</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;12-33-1&quot;</td>
<td>10</td>
</tr>
<tr>
<td>&quot;044&quot;</td>
<td>8</td>
</tr>
<tr>
<td>&quot;044-0-7-02&quot;</td>
<td>15</td>
</tr>
<tr>
<td>&quot;17-3457-ab&quot;</td>
<td>-1</td>
</tr>
<tr>
<td>&quot;Summertime&quot;</td>
<td>-1</td>
</tr>
</tbody>
</table>

**Hints:**

1. The simplest approach to this problem is to go through all characters in the string one by one, checking if the character is a digit, dash or anything else, and computing the total sum of all digits.

2. You can use method `Character.isDigit()` to check if a given character is a digit. Method `Character.isDigit()` must be passed a parameter of type `char` and it returns a boolean value. For example, if `ch` is a variable of type `char` and `ch = '6'`, `Character.isDigit(ch)` will return `true`. If `ch = 'a'`, `Character.isDigit(ch)` will return `false`.

3. You can use method `Integer.parseInt()` to convert a string containing a digit into an integer. Method `Integer.parseInt()` must be passed a parameter of type `String` and it returns a value of type `int`. For example, `Integer.parseInt("3")` returns an integer value 3. The method will generate a run-time error if the string passed as a parameter contains any non-digit characters.
public static ______ computeChecksum(______________________)
Compose a main method that prompts the user to enter a license string and uses the method computeChecksum() developed in the first part to test the license number entered by the user. The main method must print “Incorrect license format”, if method computeChecksum returns -1, otherwise it should print the value returned by method computeChecksum.
Problem 3. (45 points) Consider the definition of class Item specified below.

```java
public class Item {
    private String name;
    private double price;

    // Accessors
    public double getPrice() { return this.price; }
    public String getName() { return this.name; }
}
```

Add the following methods to the class.

a. (10 points) Public static method isValidItemName() that is passed a parameter of type String and returns true if and only if the string has format “LL-DD”, where L is any single letter, the third character is a dash (“-”) and D is any single digit.

*Hint:* Static method Character.isLetter(ch) returns true if and only if character ch is a letter, static method Character.isDigit(ch) returns true if and only if character ch is a digit.
b. (10 points) A **1 argument constructor** that is passed a parameter of type String and sets the instance variable `name` to the value of that parameter only if the parameter denotes a valid item name, i.e. has format “LL-DD”. The constructor must use method `isValidItemName()` to verify if the parameter string has the correct format.

The constructor should set the `price` instance variable to 0.

c. (5 points) Public instance method `setPrice()`, which is passed a parameter of type `double` and sets the `price` instance variable of the calling object to the value of the parameter, if the parameter is non-negative. If the passed parameter is negative, the method should print an appropriate message and leave the `price` instance variable unchanged.
d. (20 points) In the unfinished program below, add code to

- Create an object of class Item using the itemName obtained from the user. Store the created object in variable itm.
- Assign value of itemPrice to the price instance variable of the object stored in itm using the setPrice() method defined in part c of this problem.
- Compare the price of the item stored in variable itm with the value 1.5 and print “Low price” if the price of the item is below 1.5. Use the appropriate accessor method of the Item class.

```java
public class UseItem {
    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        System.out.println("Please enter the item name");
        String itemName = kb.nextLine();
        Item itm;

        System.out.println("Please enter the price of the item");
        double itemPrice = kb.nextInt();
    }
}
```
Problem 4. (30 points) This problem is based on the definitions of classes Date and Person presented below.
The first set of questions is based on class Date.

```java
public class Date {
    // declare instance variables
    private String month;
    private int day;
    private int year;

    /* Constructors*/
    public Date (){
    }
    public Date(String month, int day, int year){
        this.month = month;
        this.year = year;
        this.day = day;
    }

    // mutator
    public void setDate (String aMonth, int aDay, int aYear ){
        this.month = aMonth;
        this.day = aDay;
        this.year = aYear;
    }

    // accessor methods
    public int getYear() { return this.year;}
    public int getDay() { return this.day;}
    public String getMonth(){ return this.month;}

    /* Instance method century() -returns the century corresponding to the year of the calling object. */
    public int century(){
        int result = (this.year/100) + 1;
        return result;
    }
}
```
a. (10 points) Complete the definition of the public instance method `sameMonth()` (see above) of class `Date`. The method should be passed a parameter of class type `Date`, and should return true if and only if the calling object's `month` instance variable has the same value as the `month` instance variable of the parameter object.

b. (5 points) Assume variables `lastDayOfClass` and `cs180final` each store an object of class `Date`. Write a statement that calls instance method `sameMonth()` to check if the dates stored in `Date` variables `lastDayOfClass` and `cs180final` are from the same month.

Question c below uses definition of class `Person`. 
public class Person {
    private String name;
    private Date birthday;

    public Person (String name, Date bd){
        this.name  = name;
        this.date  = bd;
    }

    // Accessors
    public Date getBirthday() { return this.birthday; }
    public String getName(){ return this.name; }
}

c. (15 points) Write a code segment that
   • creates an object of class Person, representing Samuel Clemens, born in November 30, 1835, and stores it in the variable called p (do not forget to include the variable declaration),
   • prints out the century corresponding to the birthday of Samuel Clemens by invoking appropriate methods of the Person and Date classes. If you need additional variables, do not forget to include their declarations.