

Practice Problems: Classes and Objects (Chapters 5 and 6)

1) The Java class called `Holiday` is started below. An object of class `Holiday` represents a holiday during the year. This class has three instance variables:

- `name`, which is a `String` representing the name of the holiday
- `day`, which is an `int` representing the day of the month of the holiday
- `month`, which is a `String` representing the month the holiday is in

```
public class Holiday {
    private String name;
    private int day;
    private String month;

    // your code goes here
}
```

a) Write a constructor for the class `Holiday`, which takes a `String` representing the name, an `int` representing the day, and a `String` representing the month as its arguments, and sets the class variables to these values.

An answer:

```
public Holiday(String n, int d, String m) {
    name = n;
    day = d;
    month = m;
}
```

b) Write a method `inSameMonth`, which compares two instances of the class `Holiday`, and returns the Boolean value `true` if they have the same month, and `false` if they do not.

An answer:

```
public boolean inSameMonth(Holiday hol){
    return this.month.equals(hol.month);
}
```

c) Write a method `avgDate` which takes an array of base type `Holiday` as its argument, and returns a `double` that is the average of the `day` variables in the `Holiday` instances in the array. You may assume that the array is full (i.e. does not have any `null` entries).

An answer:

```
public static double avgDate(Holiday[] hol) {
    int sum = 0;

    for(int i = 0; i < hol.length; i++) {
        sum = sum + hol[i].day;
    }

    return ((double) sum)/hol.length;
}
```

d) Write a piece of code that creates a `Holiday` instance with the name "Independence Day", with the day "4", and with the month "July".

An answer:

```
Holiday hol = new Holiday("Independence Day", 4, "July");
```

2) The class `Movie` is started below. An instance of class `Movie` represents a film. This class has the following three class variables:

- `title`, which is a `String` representing the title of the movie
- `studio`, which is a `String` representing the studio that made the movie
- `rating`, which is a `String` representing the rating of the movie (i.e. PG-13, R, etc)

```
public class Movie {
    private String title;
    private String studio;
    private String rating;

    // your code goes here
}
```

a) Write a constructor for the class `Movie`, which takes a `String` representing the title of the movie, a `String` representing the studio, and a `String` representing the rating as its arguments, and sets the respective class variables to these values.

An answer:

```
public Movie(String t, String s, String r) {
    title = t;
}
```

```
        studio = s;
        rating = r;
    }
```

- b) Write a second constructor for the class `Movie`, which takes a `String` representing the title of the movie and a `String` representing the studio as its arguments, and sets the respective class variables to these values, while the class variable `rating` is set to "PG".

An answer:

```
public Movie (String t, String s) {
    title = t;
    studio = s;
    rating = "PG";
}
```

- c) Write a method `getPG`, which takes an array of base type `Movie` as its argument, and returns a new array of only those movies in the input array with a rating of "PG". You may assume the input array is full of `Movie` instances. The returned array need not be full.

An answer:

```
public static Movie[] getPG(Movie[] mov) {
    Movie[] pgMov = new Movie[mov.length];

    int newArrayIndex = 0;
    for (int i = 0; i < mov.length; i++) {
        if (mov[i].rating.equals("PG")) {
            pgMov[newArrayIndex] = mov[i];
            newArrayIndex++;
        }
    }
    return pgMov;
}
```

- d) Write a piece of code that creates an instance of the class `Movie` with the title "Casino Royale", the studio "Eon Productions", and the rating "PG-13".

An answer:

```
Movie mov = new Movie("Casino Royale", "Eon
                        Productions", "PG-13");
```