

CS163/164: Worksheet Week 6 – Methods Overloading

Name(s): _____

1. Analyze the classes below and write the output of the program if you execute it.

```
public class DatePrinter {
    public void datePrint(int day, int month, int year) {
        System.out.print("1");
    }
    public void datePrint(int day, String month, int year) {
        System.out.print("2");
        datePrint(day, Integer.parseInt(month), year);
    }
    public void datePrint(int month, int year) {
        System.out.print("3");
        datePrint(1, month, year);
        datePrint(1, String.valueOf(month), year);
    }
}

public class AppDatePrinter {
    public static void main(String args[]){
        DatePrinter dp = new DatePrinter();
        dp.datePrint(22,2, 2023);
        System.out.println();
        dp.datePrint(22,"2", 2023);
        System.out.println();
        ArrayList<String> months = new ArrayList<>();
        months.add("January"); months.add("February");
        dp.datePrint(months.indexOf("January"), 2023);
    }
}
```

Terminal

2. Implement the methods as specified.

```
import java.util.ArrayList;
public class Contact {
    private long phone;      private String name;

    public Contact(String name) {
        this.name = name;      phone = 0;
    }
    public Contact(String name, String phone){
        //to do - initialize name and phone, phone will use the appropriated setPhone
        method

    }
    public Contact(String name, long phone){
        //to do - initialize name and phone, phone will use the appropriated setPhone
        method

    }

    public String getFormattedPhone() {
        return String.format("(%d) %d-%d", getAreaCode(), getPrefix(), getNumber());
    }
}
```

CS163/164: Worksheet Week 6 – Methods Overloading

Name(s): _____

```
public int getPrefix() {      long tmp = phone / 10000;      return (int) (tmp % 1000);
}
public int getAreaCode() {    long tmp = phone / 100000000;    return (int) (tmp %
1000); }
public int getNumber() {      return (int) phone % 10000; }
public long getPhone() {      return phone; }

public void setPhone(long phone) {
    // to do - just initialize the instance variable with the parameter
}
public void setPhone(String phone) {
    // to do

    //transform the String into a long number

    //remove all characters that are not numbers

    //Long.parseLong(string)

    //call overloaded setPhone
}
public String toString() {
    return String.format("Name: %s, phone: %s", name, getFormattedPhone());
}

public static void main(String[] args) {
    ArrayList<Contact> advisors = new ArrayList<>();
    Contact bess = new Contact("Bess");
    bess.setPhone(9704915944L); // the "L" at the end is how we tell java it is a
long number
    advisors.add(new Contact("Gabbi","(970) 491-3739"));
    advisors.add(new Contact("Tran"));
    advisors.add(new Contact("Heidi"));
    advisors.add(bess); // just doing this so you can see adding other ways to add
objects to ArrayLists
    for(Contact advisor:advisors) {
        if(advisor.getPhone() > 0) {
            System.out.println(advisor);
        } //end if
    } //end for
} // end main
} //end class
```

3. Analyze the class Rectangle and implement the appropriated methods.

```
public class Rectangle {
    private double height;
    private double width;

    public Rectangle(double height, double width){
        setHeight(height);
        this.setWidth(width);
    }
    public void setHeight(double height){
        this.height = height ;
    }
    public void setWidth(double width){
        this.width = width;
    }
    public double getHeight(){
```

CS163/164: Worksheet Week 6 – Methods Overloading

Name(s): _____

```
        return height;
    }
    public double getWidth(){
        return width;
    }
}
```

- Rectangle Class
 - Write a method that calculates the area of a rectangle.
 - Write a method that receives a Rectangle as a parameter, compare the areas and returns a String with the height, width, and area of the biggest rectangle. If it is the same area, return "Same".
- RectangleApplication Class
 - Write the method readDouble: receives a Scanner as a parameter, read, and return a double value;
 - Write a main method that creates two Rectangle objects (use readDouble to read the values to create the objects), prints both objects height and width, calls the methods that return the biggest rectangle.