

---

# University of Colorado at Colorado Springs

## Home Work Assignment 2 Out 09/23/2019, Due 10/07/2019

---

### 1 Computing Surface Area of a Sphere (Validating Input) (10 pts code + 5 pts pseudocode = 15 pts)

Write a Java program `ValidateInput.java` that asks for a number to be entered by the user from the terminal. The program prints out the surface area of a sphere with this number as its radius. Note that with a radius of  $r$ , the surface of a sphere is  $4\pi r^2$ .

This program requires that  $r$  can only be a positive number. If the user enters a negative number or 0, your program should print a message “Please enter a positive number. Good bye!” and exit. To exit a Java program, you can call `System.exit(1);` (just like the way you call `System.out.println(‘...’);`). Any statements after `System.exit(1);` will not be executed.

You can enter a number with any number of digits after the decimal point, but make sure when you print the surface area, print it with **3 decimal points**. Two separate runs may look like the following.

```
Enter the radius: -1
Please enter a positive number. Good bye!
```

```
Enter the radius: 2.125
Surface of the sphere: 56.743
```

### 2 Finding the Day of the Week (25 pts code + 5 pts pseudocode = 30 pts)

Write a program `DayOfWeek.java` that prompts the user to enter a number from 0 to 6, and displays the name of the day (e.g., 0 for Sunday, 1 for Monday, etc.) Two separate runs may look like the following.

```
Enter a number from 0 to 6: 7
Please make sure the number is between 0 and 6. Good bye!
```

```
Enter a number from 0 to 6: 5
Today is Friday!
```

Please use a `switch` statement to implement this. The program should handle incorrect user input (i.e., the number that is not between 0 and 6). To exit a program upon incorrect input, please refer to Question 1. We can assume that the user input is always an integer.

### 3 Airline Reservation (40 pts code + 5 pts pseudocode = 45 pts)

Create a very simple airline reservation program `AirReservation.java` that allows the user to select a destination city and the number of checked bags. The program will report the total cost of the trip (air fare + checked bag fee). Assume that only one person is travelling, and the airline flies only from Denver to three cities and the fares are fixed as the following:

Flight 1: Sydney, Australia, fare: \$1750.00

Flight 2: London, England, fare: \$1300.00

Flight 3: Frankfurt, Germany, fare: \$1450.00

The airline also allows from 0 up to 2 checked bags, with \$20.00 per bag.

Please write a program that prompts the user for: the city she would like to travel to (using integer 1–3 to select one of the flights above), and the number of checked bags she will be traveling with. Finally, the program displays the overall cost of the trip (air fare + checked bag fee).

The code must handle user validation for: (1) flight selection – if an invalid flight number is entered, **MUST** print a message and `EXIT`, and (2) checked bags – if invalid baggage numbers are entered, **MUST** print a message and `EXIT`. To exit a program, please refer to Question 1. You may use either a `switch` or `if` statement to implement this. A normal run may look like the following:

```
Where would you like to travel to? Select option 1, 2, or 3: 3
How many bags do you want to check ($20 per bag)? 2
Your flight's total cost is: $1490.00
```

Some error handling output may look like the following:

```
Where would you like to travel to? Select option 1, 2, or 3: 0
```

```
0 is not a valid flight number.  
Program can't deal with flight 0! Good bye!
```

```
Where would you like to travel to? Select option 1, 2, or 3: 2  
How many bags do you want to check ($20 per bag)? 9  
You may only have 0 to 2 checked bags!  
Program can't deal with 9 checked bags! Good bye!
```

## Submission

Please save your programs in three Java files, each containing **pseudocode**. You may include your pseudocode in a block comment using `/* ... */`. **10 pts are given to your coding style** (comments – header and in-code comments: up to 4 pts, naming conventions: up to 3 pts, proper indentation/spacing: up to 3 pts). We will run each program several times with our input and verify that the results are correct.

Please place your files in a folder called **hw2-firstname-lastname** and zip it. The zipped file should be named **hw2-firstname-lastname.zip**. Please submit the zipped file to Canvas by the due date.