

CS-420-1: Object-Oriented Design
Fall 2019
Northeastern Illinois University
Homework #6: Due Thursday, 10/24/19 at 5:00 p.m.
Lambdas

Part I: Short/Long Answer Questions - Instructions

- All of your responses should be typed out and submitted in **.pdf format**. Upload the .pdf file to D2L.
- **If your assignment is not submitted in .pdf format you will receive a zero for the assignment (because you did not read the instructions!!).**
- For each question, make sure that you put the question number first, followed by your answer - using complete sentences and grammar when asked to provide a justification.
- Upload the .pdf file **separately** from the python project zip file.
- **A note on plagiarism:** Your answers must be your own work. Significant plagiarism of answers will result in a zero for the entire homework grade. TurnItIn will be used to check for plagiarism (and you will see your plagiarism score and which text blocks are plagiarized when you upload the file).

When specified, justify your choice:

1. Fill in the blanks: The _____ functional interface does not take any inputs, while the _____ functional interface does not return any data.
 - (a) IntConsumer, LongSupplier
 - (b) IntSupplier, Function
 - (c) Supplier, DoubleConsumer
 - (d) UnaryOperator, Consumer
2. What is the output of the following application (options on next page)? Explain your choice.

```
import java.util.function.*;

class Tourist {
    public Tourist(double distance) {
        this.distance = distance;
    }
    public double distance;
}

public class Lifeguard {
    private void saveLife(Predicate<Tourist> canSave, Tourist tourist) {
        System.out.print(canSave.test(tourist) ? "Saved" : "Too far"); // y1
    }
    public final static void main(String... sand) {
        new Lifeguard().saveLife(s -> s.distance < 4, new Tourist(2)); // y2
    }
}
```

- (a) Saved
 - (b) Too far
 - (c) The code does not compile because of line y1.
 - (d) The code does not compile because of line y2.
3. Assuming the proper generic types are used, which lambda expression cannot be assigned to a ToDoubleBiFunction functional interface reference? Explain your choice.
- (a) (Integer a, Double b) -> {int c; return b;}
 - (b) (h,i) -> (long)h
 - (c) (String u, Object v) -> u.length() + v.length()
 - (d) (x,y) -> {int z = 2; return y/z;}
4. What is the output of the following application? Explain your choice.

```
import java.util.function.*;

public class TicketTaker {
    private static int AT_CAPACITY = 100;
    public int takeTicket(int currentCount, IntUnaryOperator<Integer> counter) {
        return counter.applyAsInt(currentCount);
    }

    public static void main(String...theater) {
        final TicketTaker bob = new TicketTaker();
        final int oldCount = 50;
        final int newCount = bob.takeTicket(oldCount,t -> {
            if(t>AT_CAPACITY) {
                throw new RuntimeException("Sorry, max has been reached");
            }
            return t+1;
        });
        System.out.print(newCount);
    }
}
```

- (a) 51
 - (b) The code does not compile because of lambda expression.
 - (c) The code does not compile for a different reason.
 - (d) The code compiles but prints an exception at runtime.
5. List the Java API functional interfaces that return a primitive value.

Part II: Coding - Getting Started

- Download the files provided for you from the course website and unzip them into a folder.
- Rename the folder to homework6.
- With IntelliJ (v2019.2.3RC or higher), choose `Import Project` and navigate to the homework6 folder.
- Choose `Import model from external model` and select `Gradle` and click `Finish`.
- Go to `Preferences` → `Build, Execution and Deployment` → `Build Tools` → `Gradle`.
- Make sure that the checkbox for `Automatically import this project on changes in build script files` is checked and that both the `Build` and `run using:` and `Run tests using:` drop downs are set to `Gradle`. Change the `Use Gradle` from `'gradle-wrapper.properties'` file to `'wrapper'` task in `Gradle build script`. Make sure that the `Gradle JVM` is set to `Java 11`!
- Also go to `File` → `Project Structure/Project Settings` → `Project` and make sure that the `JDK` is set to `Java 11`.

Problem 1:

1. Create a class named `InputProperties`.
2. Create a static method named `isOdd`. The `isOdd` method should return a lambda function that takes an integer and returns `true` if the integer is even and `false` otherwise. The lambda function should be no more than one line. What should the return type of the method be? Hint: Take a look at the Java API functional interfaces! Uncomment the first two tests in the `TestInputProperties` class.
3. Create a static method named `isPrime`. The `isPrime` method should return a lambda function that takes an integer and returns `true` if the integer is a prime number and `false` otherwise. Challenge: Can you write the lambda function in one line? What should the return type of the method be? Hint: Take a look at the Java API functional interfaces! Uncomment the next two tests in the `TestInputProperties` class.
4. Create a static method named `isPalindrome`. The `isPalindrome` method should return a lambda function that takes an integer and returns `true` if the integer is a palindrome (i.e. reads the same way forward and backward) and `false` otherwise. Challenge: Can you write the lambda function in one line? What should the return type of the method be? Hint: Take a look at the Java API functional interfaces! Uncomment the next two tests in the `TestInputProperties` class.
5. Create a static method named `check` that takes an integer and a parameter that is the type of functional interface returned by the methods that you just created. In this method, call the abstract method of the functional interface parameter and return the appropriate type. Uncomment the final test in the `TestInputProperties` class.

6. Create the main method in the `InputProperties` and demo the use of the above methods that you created (i.e. you should show how the `check` method interacts with each of the other methods to produce output).

Submitting your homework to D2L

- In IntelliJ, under `File`, choose "Export to Zip File".
Use the name of `homework6.zip`.
- Submit this file to D2L.