CS571 HW2

February 18, 2024

Total Score : 45

Instructions

This homework consists of 3 theoretical questions and 1 programming question

Please provide your answers to all theoretical questions in a PDF format.

- Upload the PDF containing your answers to Gradescope.
- Submit the programming source tarball on Brightspace under the appropriate assignment.

1 Inheritance and Subtyping Total Score: 15

Consider the following Java program:

```java
class Point {
  int x; int y;

  Point(int x_, int y_) { x = x_; y = y_; }
  Number similarity(Point other) {
    // Computes the L2 norm between this point and "other",
    // and returns the result.
    . . .
  }
}

class ColorPoint extends Point {
  int color;
  float color_factor = 0.3f;
  ColorPoint(int x_, int y_, int color_) { super(x_, y_); color = color_; }

  Float similarity(ColorPoint other) {
    // Computes the L2 norm between this point and "other",
    // including color in the distance computation. The
    // tradeoff between pointwise distance and color
    // distance is controlled with color_factor.
    . . .
  }
}

public class App {
  public static void main(String[] args) {
    Point p = new Point(2,1);
    ColorPoint q = new ColorPoint(3,5,127);

    System.out.println(p.similarity(q)); // 1st line of output
    System.out.println(q.similarity(p)); // 2nd line of output
  }
}
```

1. Does this program a) produce an error at compile time, b) produce an error at run time, c) compile and run successfully, with both lines of output producing the same result, or d) compile and run successfully, with each line of output producing a different result. If it produces an error, describe the error.

2. If the program produces an error, describe a change you could make to the program that fixes the error and causes it to execute successfully, and describe the output of the successful execution. If the program does not produce an error, describe a change you could make to change the program's output behavior (i.e. if the program produces a different output on each line, how could you make both output lines the same and vice versa).

3. Imagine we created a new language called Java++ that is exactly like Java, but uses only method names to perform method calls (as opposed to Java which uses both method names and parameter types). Furthermore, imagine that Java++ supports contravariant subtyping for method parameters in the same way that Java supports covariant subtyping for method return values. How would your answers to parts 1. and 2. of this question change if this program were written in Java++ instead of Java?

2 Stack-Oriented Programming Total Score: 10

The following programs are Java bytecode programs. Do these programs perform the same computation or different computations (i.e. given any initial state of the registers accessed by `iload` and `istore` , do the programs produce the same final state in those registers)? If they are the same, prove they are the same. If they are different, provide an example input state that produces a different output state in each program.

```
iconst 0
istore w
iload y
iconst 5
iload a
iload x
imul
iload b
iadd
istore y
imul
iload c
iadd
istore z
```

```
iload y
istore w
iload b
iload a
iload x
imul
iadd
istore y
iconst 5
iload w
imul
iload c
iadd
istore z
iconst 0
istore w
```

3 Polymorphism Total Score: 10

The following program contains the class `SecurePass`, which is supposed to maintain the privacy and security of its field `password`. The class `PasswordClient` is a client that uses this password system.

1. Write an implementation of the method `run_client` to extract the supposedly private field `password` from the object `p`.

2. How would your answer to part 1. change if `password` were declared `private`?

```java
class PasswordClient {
    public void run_client(SecurePass p) {
       // try to access the password string here and spring it
    }
}

interface SecurePass {
    boolean checkPassword(String password);
}

class CheckPassword implements SecurePass {
    String password;

    public CheckPassword(String password) {
        this.password = password;
    }

    @Override
    public boolean checkPassword(String inputPassword) {
        return password.equals(inputPassword);
    }
}

class Main {
    public static void main(String[] args) {
        CheckPassword checker = new CheckPassword("mySecretPassword");
        SecurePass securePass = new CheckPassword("anotherPassword");

        PasswordClient client = new PasswordClient();
        client.run_client(securePass);
    }
}
```

4 Programming Assignment Total Score: 10

source code gitlab link : `https://gitlab.com/eatkinson/cs-571-programming-assignments.git`

4.1 Part 1

Note - source code can be downloaded form the gitlab link provided and submit the completed source code tarball on to brightspace

implement the eval function for all non-abstract subclasses of Expr in Expr.java file

4.2 Part 2

Note - you are not allowed to use the instanceof operator and also not allowed to do down-casting

Implement the "equals" function for all non-abstract subclasses of Expr, which checks "structural equality" between the "this" expression and the input expression "other". Structural equality means comparing two abstract syntax trees to determine if they represent the same syntactic structure. It requires you to traverse the tree and determine if a) all floating-point values at the leaves are the same, b) all operations in the middle of the tree are the same, and c) the operations are arranged in the same manner between the two trees. Tests for this behavior are available in "Main.java".

For this part, you may not use downcasting, "instanceof", or any other Java tools to access the runtime type of a class such as the reflection library. Use the visitor pattern instead! Note that implementing the visitor pattern will require you to extend the abstract base class Expr with additional methods.

5 Additional Resources

5.1 Clone the gitlab Repository

1. Open a terminal or command prompt on your local machine(recommended to clone on to remote server and work there).
2. Use the following command to clone the repository to your local machine:

   ```
   git clone repository-url
   ```

   Replace `repository-url` with the URL of the GitLab repository.

6 Make Necessary Changes

1. Navigate to the cloned repository directory on your local machine.
2. Make the necessary changes to the assignment files using your preferred text editor or IDE.

6.1 Create a Tarball of the Assignment

1. Once you've made the changes, navigate to the parent directory of the repository.
2. Use the following command to create a tarball of the modified assignment files:

   ```
   tar -czvf assignment.tar.gz repository-name
   ```

   Replace `repository-name` with the name of the cloned repository.

6.2 Submit the Tarball on Brightspace

1. Log in to Brightspace and navigate to the specific assignment or assessment.
2. Look for an option to upload files or submit an assignment.
3. Click on the "Upload" button or similar option.
4. Locate the tarball file (`assignment.tar.gz`) on your computer and select it for upload.
5. Submit the assignment following the instructions provided by your instructor.