CSE216 – Programming Abstractions
Recitation 2

Section 1:

Go through the online article that discusses Memory Layout of C Programs.

https://www.geeksforgeeks.org/memory-layout-of-c-program/

Submit the programs you wrote and their output on blackboard.

For more information about how to understand stack and heap memory usage, follow the online article available at:

https://medium.freecodecamp.org/understand-your-programs-memory-92431fa8c6b

Section 2:

Download the CSE216Rec.zip project on your computer and open with NetBeans. We will learn about polymorphism in Java through various demos.

Polymorphism:

The word 'polymorphism' literally means 'a state of having many shapes' or 'the capacity to take on different forms'. When applied to object oriented programming languages like Java, it describes a language's ability to process objects of various types and classes through a single, uniform interface.

Demo: PolymorphismDemo.java

Polymorphism in Java has two types: Compile time polymorphism (static binding) and Runtime polymorphism (dynamic binding).

Method overloading is an example of static polymorphism.

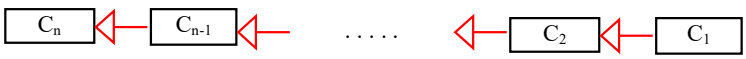
Demo: DisplayOverloading.java

Method overriding is an example of dynamic polymorphism.

Demo: OverridingDemo.java

Dynamic binding: Suppose an object o is an instance of classes C_1, C_2, ..., C_{n-1}, and C_n

- C_1 is a subclass of C_2, C_2 is a subclass of C_3, ..., and C_{n-1} is a subclass of C_n
- C_n is the most general class, and C_1 is the most specific class
- If o invokes a method p, the JVM searches the implementation for the method p in C_1, C_2, ..., C_{n-1} and C_n, in this order, until it is found, the search stops and the first-found implementation is invoked.



Since o is an instance of C_1, o is also an instance of C_2, C_3, . . . , C_{n-1}, and C_n

Demo: DynamicBinding.java

Generic programming: Java Generic methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods, or with a single class declaration, a set of related types, respectively.

Demos: GenericMethodTest.java

BoundedTypes.java

GenericBox.java