WEEKLY-EXERCISE - 08

ICS 365-51 **Metropolitan State University/MN**

Week 9 **Due 11:59pm, Sunday, Oct. 23rd, 2022** **Fall 2022**

Name:__________________Pong Lee__________________ **Score: __________________**

Please complete both Parts I and II and then upload the results to D2L under the dropbox for Weekly Exercise 08 before the deadline (total 20 points).

Part I: Based on the discussion in Lecture 8, please either **bold** or highlight your answers below, only one answer per question. (1 point each, total 10 points)

1. Based on the discussion in Chapter 10, general semantics of subprogram returns will consider all the following issues *except*?

   A) stack-dynamic allocation of local variables;

   B) returning control to the caller;

   C) restoring the execution status;

   D) deallocation of stack-dynamic locals.

2. Based on the discussion in Chapter 10, which of the following statements is NOT true?

   A) Blocks are user-specified local scopes for variables;

   B) To restore the execution status of the caller is done by the caller itself;

   C) A static chain is a chain of static links that connects certain activation record instances;

   D) In the discussion of Chapter 10, EP stands for Environment Pointer.

3. Based on the discussion in Figure 10.7 (page 428 of your textbook or Slide 21 for Chapter 10), which of the followings is the value of parameter *n* in the third recursive call to function *factorial* with an initial value of *n* as *5* ?

   A) 1

   B) 2

   C) 3

   D) 4

4. Based on the discussion in Chapter 10, general semantics of calls to a subprogram will consider all the following issues *except*?

   A) Saving the execution status of calling program;

   B) Parameter passing methods;

   C) The spelling of the keywords used in the subprogram;

   D) Transferring of control and arranging for the return.

5. Based on the situation presented in Figure 10.12 (page 440 of your textbook or Slide 36 for Chapter 10), which of the followings is true after `sub3` calls `sub1` for the program units of the variable declaration?

   A) For variable v: main, sub1, sub1, sub3;
   B) For variable x: sub2, sub3, sub1;

   C) For variable z: sub3, sub3;
   D) For variable w: sub1, sub1, sub2, sub1.

6. Based on the discussion in Chapter 11, which of the followings is implemented in *Java* but not in *C++?*

A) The constructor’s name is the same as the class name;

B) All of the class instances of a class share a single copy of the member functions;

C) Implicit garbage collection of all objects;

D) Each instance of a class has its own copy of the class data members.

7. Based on the discussion in Chapter 11, Which of the followings is not one of the design issues for Abstract Data Type (ADT)?

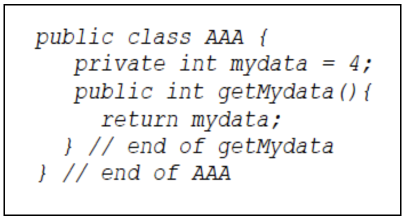
A) What access controls are provided?

B) Is the specification of the type physically separated from its implementation?

C) Can abstract types be parameterized?

D) How much CPU time is needed to achieve ADT?

8. If class AAA is defined as follow and aaa is an object of AAA,



```
public class AAA {
   private int mydata = 4;
   public int getMydata(){
     return mydata;
  } // end of getMydata
} // end of AAA
```

which of the following statements is correct?

A) `System.out.println(“mydata = “ + getMydata());`

B) `System.out.println(“mydata = “ + aaa.mydata);`

C) `System.out.println(“mydata = “ + aaa.getMydata());`

D) `System.out.println(“mydata = “ + AAA.getMydata());`

9. Which of the following statements is *unlikely* a feature of Constructor in Java?

A) It can have parameters;

B) It must have a return value;

C) It is a special method used to initialize an object;

D) It has the same name as the class itself;

E) None of above.

10. Based on the discussion in Chapter 11, which of the following statements is not true?

A) Nearly all programming languages support data abstraction with subprograms;

B) All user-defined types in Java are classes;

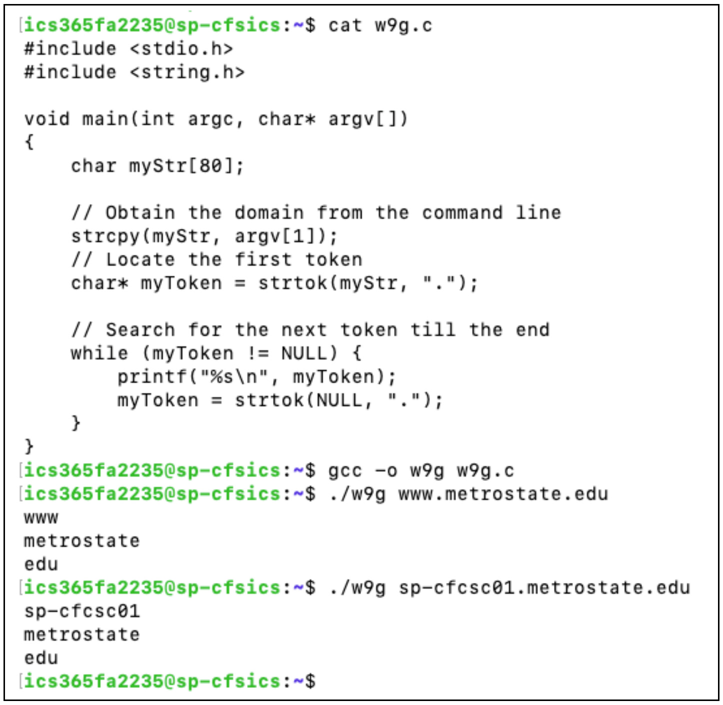
C) An abstraction is a view or representation of an entity that includes only the most significant attributes;

D) An encapsulation is defined as a collection of logically related code and data where each of which can be compiled without recompilation of the rest of the program.

Part II: Please study the lecture slides and handouts covered this week to complete the following tasks: (Total 10 points)

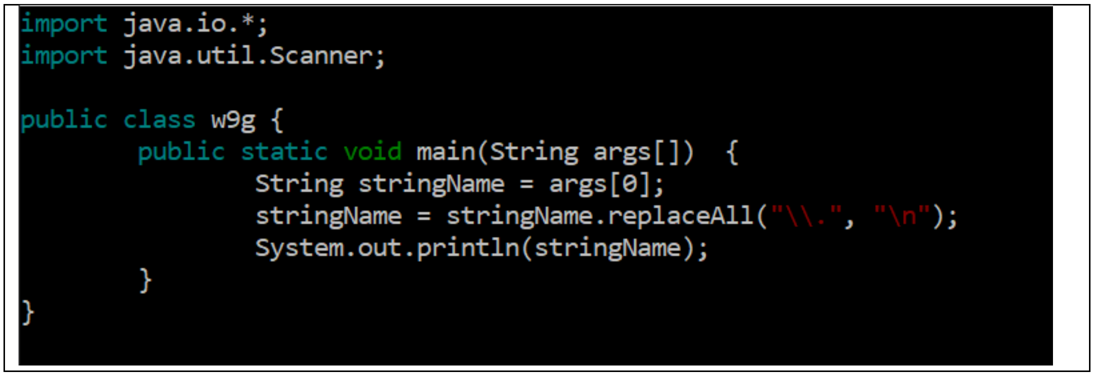
Given a *C* program as shown below, please write similar programs in Java and Python on our Linux server, *sp-cfcsc01.metrostate.edu*. Please "*cat*" your programs before either compiling and executing or executing it with the testing cases provided, and then include the corresponding screenshots below: (10 points)

A *C* program that reads a domain name from the command line and then split it into tokens and its executions with two testing cases are provided below :

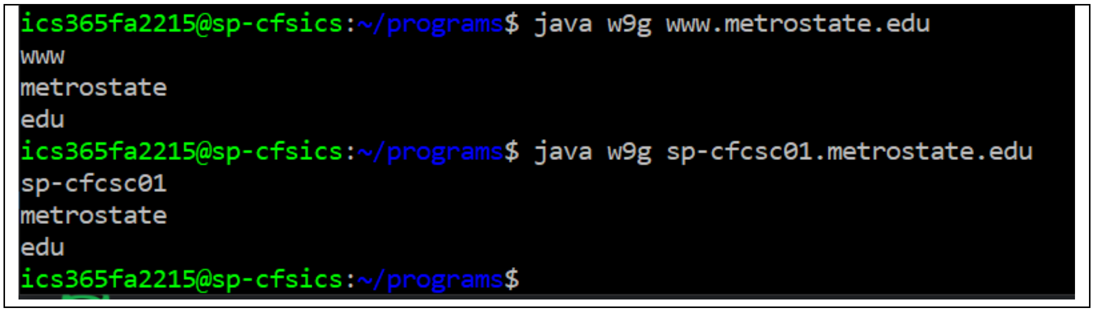


```
ics365fa2235@sp-cfsics:~$ cat w9g.c
#include <stdio.h>
#include <string.h>

void main(int argc, char* argv[])
{
    char myStr[80];

    // Obtain the domain from the command line
    strcpy(myStr, argv[1]);
    // Locate the first token
    char* myToken = strtok(myStr, ".");

    // Search for the next token till the end
    while (myToken != NULL) {
        printf("%s\n", myToken);
        myToken = strtok(NULL, ".");
    }
}
ics365fa2235@sp-cfsics:~$ gcc -o w9g w9g.c
ics365fa2235@sp-cfsics:~$ ./w9g www.metrostate.edu
www
metrostate
edu
ics365fa2235@sp-cfsics:~$ ./w9g sp-cfcsc01.metrostate.edu
sp-cfcsc01
metrostate
edu
ics365fa2235@sp-cfsics:~$
```

2.1) Please provide the screenshot of a similar program in **Java** with its execution on the two testing cases below (5 points):

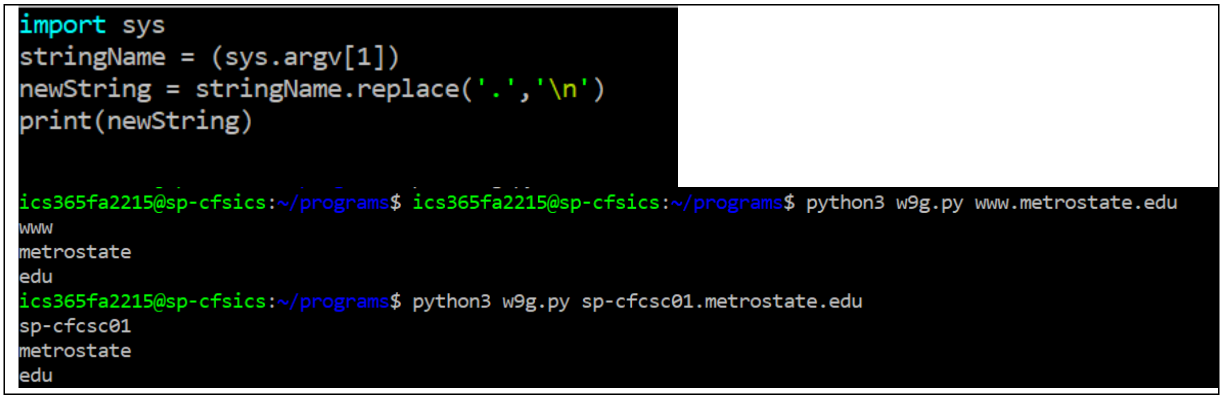


```
import java.io.*;
import java.util.Scanner;

public class w9g {
        public static void main(String args[])  {
                String stringName = args[0];
                stringName = stringName.replaceAll("\\.", "\n");
                System.out.println(stringName);
        }
}
```



```
ics365fa2215@sp-cfsics:~/programs$ java w9g www.metrostate.edu
www
metrostate
edu
ics365fa2215@sp-cfsics:~/programs$ java w9g sp-cfcsc01.metrostate.edu
sp-cfcsc01
metrostate
edu
ics365fa2215@sp-cfsics:~/programs$
```

2.2) Please provide the screenshot of a similar program in **Python** with its execution on the two testing cases below (5 points):



```
import sys
stringName = (sys.argv[1])
newString = stringName.replace('.','\n')
print(newString)
```

```
ics365fa2215@sp-cfsics:~/programs$ ics365fa2215@sp-cfsics:~/programs$ python3 w9g.py www.metrostate.edu
www
metrostate
edu
ics365fa2215@sp-cfsics:~/programs$ python3 w9g.py sp-cfcsc01.metrostate.edu
sp-cfcsc01
metrostate
edu
```