WEEKLY-EXERCISE - 09

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

Week 11 **Due 11:59pm, Sunday, Nov. 6th, 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 09 before the deadline (total 20 points).

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

1. According to Table 12.1, provided on page 518 of the textbook, which of the following statements is not true?

A) Method binding can be either dynamic or static in Java while all method bindings in Ruby are dynamic;

B) All data are objects in SMALLTALK;

C) Constructors can be implicitly called in all 5 programming languages discussed in Table 12.1;

D) C++ is the only language that supports both single and multiple inheritance.

2. Based on the discussion in Chapter 12, which of the following programming languages is a pure Object-Oriented-Programming language?

A) C++;

B) C#;

C) Java;

D) Ruby.

3. The behavior of an object in Java is defined by

A) a group of methods.

B) a set of properties.

C) a list of classes.

D) a collection of data types.

4. Based on the discussion in Chapter 12, which of the followings is not one of the three major language features in object-oriented programming?

A) Abstract Data Types;

B) Inheritance;

C) Polymorphism;

D) Functions.

5. A method that is associated with an individual object in Java is called ___________.

A) a static method.

B) a class method.

C) an instance method.

D) a block method.

6. Based on the discussion in Chapter 13, which of the following statements is true regarding the evaluation of semaphores?

A) The program will deadlock if the wait of `fullspots` is left out;

B) The buffer will overflow if the release of `access` is left out;

C) The buffer will overflow if the wait of `access` is left out;

D) The program will deadlock if the release of `access` is left out.

7. Based on the discussion in Chapter 13, task communication can be provided by

A) message passing;
B) shared nonlocal variables;
C) parameters;
D) all of above.

8. Based on the discussion in Chapter 13, synchronization can be provided by all the methods below except ___________________________________.

A) Message Passing.
B) Monitors.
C) Remote Procedure Call.
D) Semaphores.

9. Based on the discussion in Chapter 13, which of the followings is not one of the design issues for concurrency?

A) How and when tasks start and end execution;
B) How and when are tasks created;
C) How to define a variable;
D) Competition and cooperation synchronization.

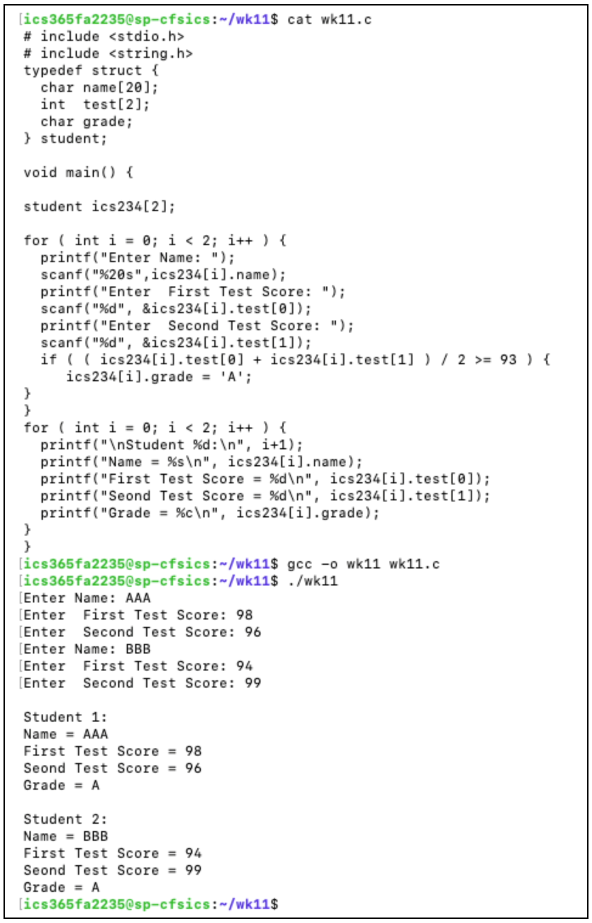
10. Based on the discussion in Chapter 13, which of the following statements is not true?

A) Scheduler is a program that maps task execution onto available processors;
B) A task that is blocked by input/output can go back to the Running state directly;
C) Cooperation synchronization refers to the situation where task A must wait for task B to complete some specific activity before task A can continue its execution;
D) Lightweight tasks all run in the same address space.

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

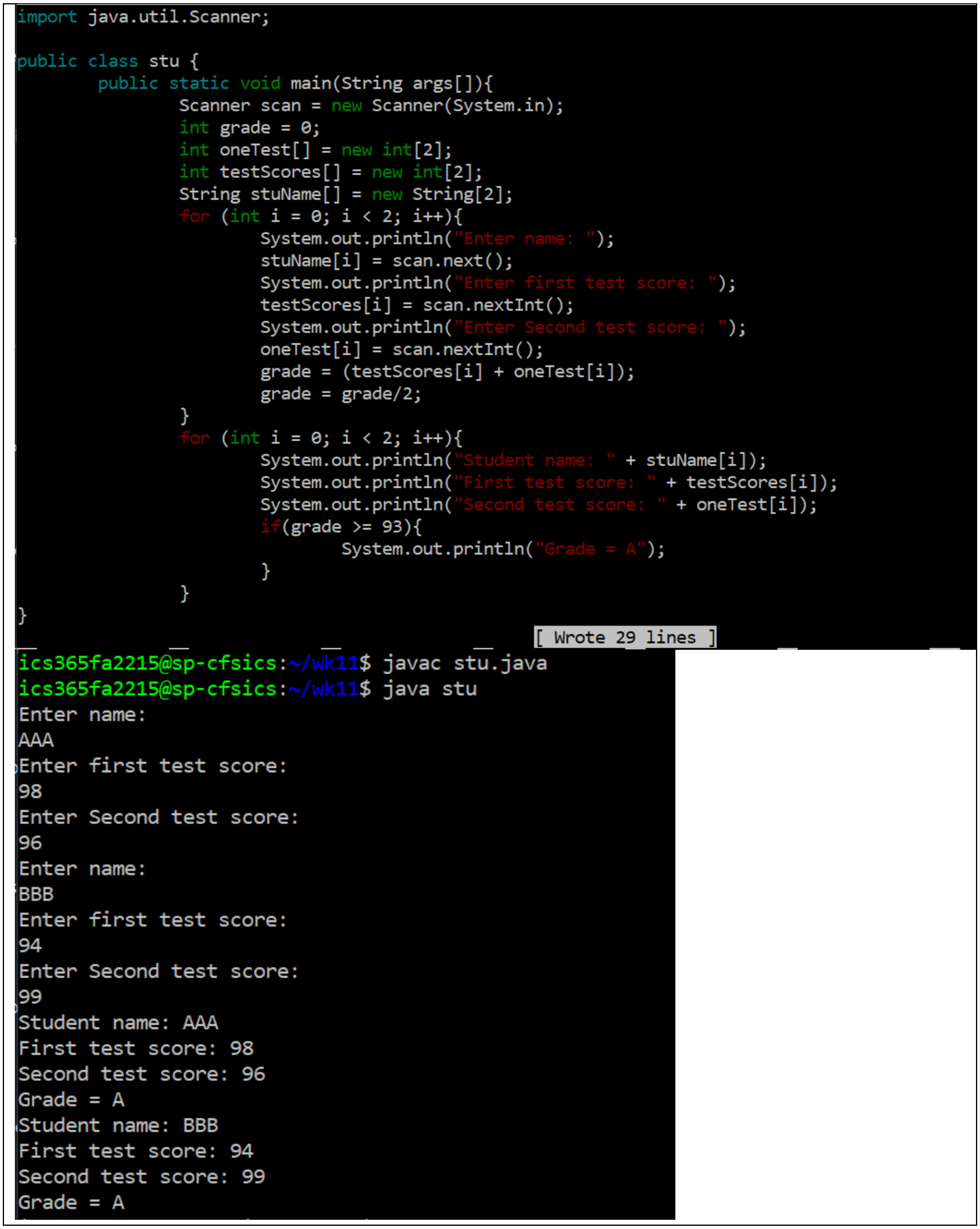
Given a *C* program as shown below, please write similar programs in Java and Python on our Linux server, *sp-cfcsc01.metrostate.edu*. Please "*cat*" your programs before either compiling and executing or executing it with the case provided, and then include the corresponding screenshots below: (10 points)

A *C* program we discussed in Handout A is provided below:



```
[ics365fa2235@sp-cfsics:~/wk11$ cat wk11.c
 # include <stdio.h>
 # include <string.h>
 typedef struct {
   char name[20];
   int  test[2];
   char grade;
 } student;

 void main() {

 student ics234[2];

 for ( int i = 0; i < 2; i++ ) {
   printf("Enter Name: ");
   scanf("%20s",ics234[i].name);
   printf("Enter  First Test Score: ");
   scanf("%d", &ics234[i].test[0]);
   printf("Enter  Second Test Score: ");
   scanf("%d", &ics234[i].test[1]);
   if ( ( ics234[i].test[0] + ics234[i].test[1] ) / 2 >= 93 ) {
      ics234[i].grade = 'A';
 }
 }
 for ( int i = 0; i < 2; i++ ) {
   printf("\nStudent %d:\n", i+1);
   printf("Name = %s\n", ics234[i].name);
   printf("First Test Score = %d\n", ics234[i].test[0]);
   printf("Seond Test Score = %d\n", ics234[i].test[1]);
   printf("Grade = %c\n", ics234[i].grade);
 }
 }
[ics365fa2235@sp-cfsics:~/wk11$ gcc -o wk11 wk11.c
[ics365fa2235@sp-cfsics:~/wk11$ ./wk11
[Enter Name: AAA
[Enter  First Test Score: 98
[Enter  Second Test Score: 96
[Enter Name: BBB
[Enter  First Test Score: 94
[Enter  Second Test Score: 99

 Student 1:
 Name = AAA
 First Test Score = 98
 Seond Test Score = 96
 Grade = A

 Student 2:
 Name = BBB
 First Test Score = 94
 Seond Test Score = 99
 Grade = A
[ics365fa2235@sp-cfsics:~/wk11$
```

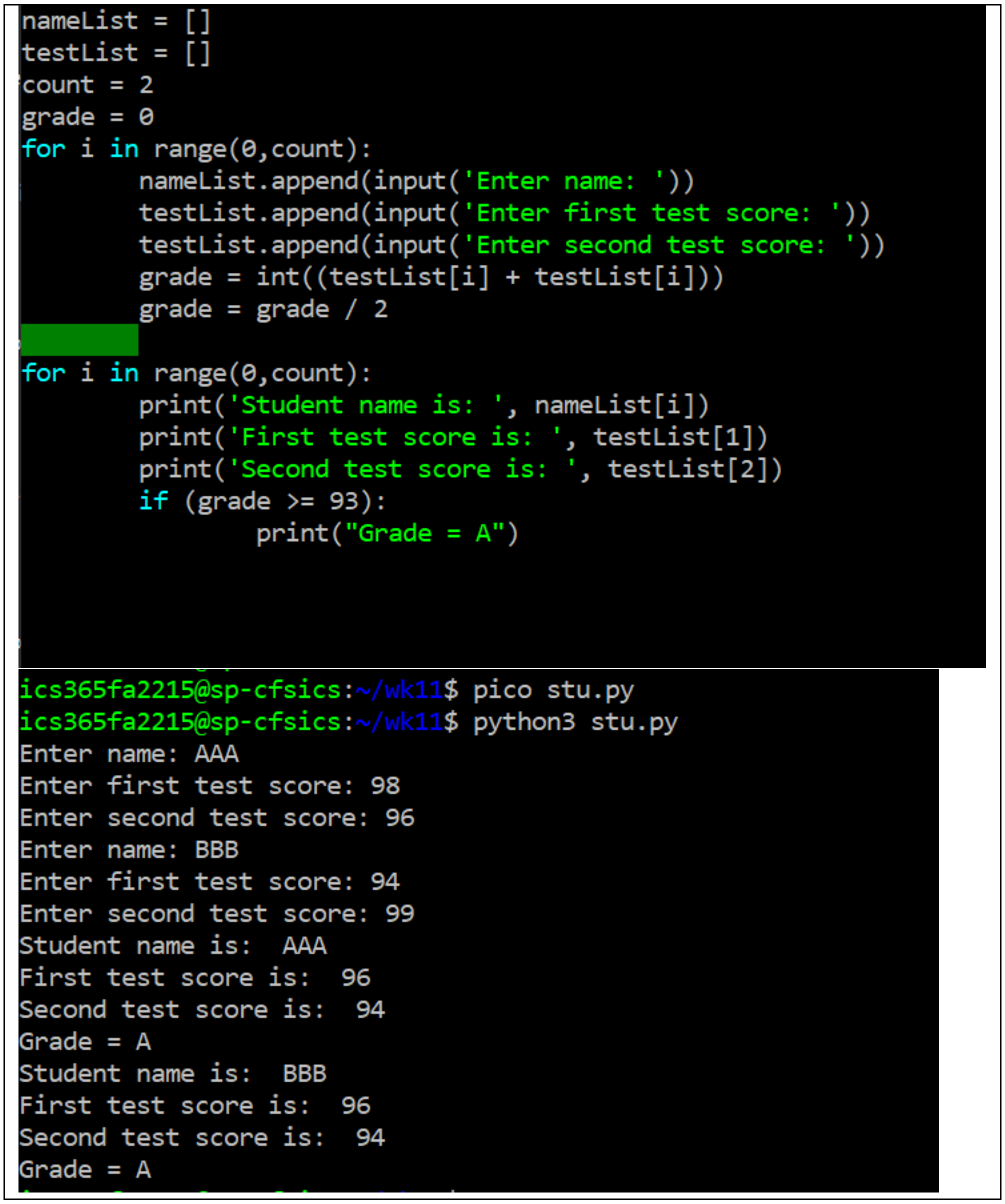
2.1) Please provide the screenshot of a similar program in **Java** with its execution on the testing case below (5 points):



```
import java.util.Scanner;

public class stu {
        public static void main(String args[]){
                Scanner scan = new Scanner(System.in);
                int grade = 0;
                int oneTest[] = new int[2];
                int testScores[] = new int[2];
                String stuName[] = new String[2];
                for (int i = 0; i < 2; i++){
                        System.out.println("Enter name: ");
                        stuName[i] = scan.next();
                        System.out.println("Enter first test score: ");
                        testScores[i] = scan.nextInt();
                        System.out.println("Enter Second test score: ");
                        oneTest[i] = scan.nextInt();
                        grade = (testScores[i] + oneTest[i]);
                        grade = grade/2;
                }
                for (int i = 0; i < 2; i++){
                        System.out.println("Student name: " + stuName[i]);
                        System.out.println("First test score: " + testScores[i]);
                        System.out.println("Second test score: " + oneTest[i]);
                        if(grade >= 93){
                                System.out.println("Grade = A");
                        }
                }
}
                                                    [ Wrote 29 lines ]
```

```
ics365fa2215@sp-cfsics:~/wk11$ javac stu.java
ics365fa2215@sp-cfsics:~/wk11$ java stu
Enter name:
AAA
Enter first test score:
98
Enter Second test score:
96
Enter name:
BBB
Enter first test score:
94
Enter Second test score:
99
Student name: AAA
First test score: 98
Second test score: 96
Grade = A
Student name: BBB
First test score: 94
Second test score: 99
Grade = A
```

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



```
nameList = []
testList = []
count = 2
grade = 0
for i in range(0,count):
        nameList.append(input('Enter name: '))
        testList.append(input('Enter first test score: '))
        testList.append(input('Enter second test score: '))
        grade = int((testList[i] + testList[i]))
        grade = grade / 2

for i in range(0,count):
        print('Student name is: ', nameList[i])
        print('First test score is: ', testList[1])
        print('Second test score is: ', testList[2])
        if (grade >= 93):
                print("Grade = A")
```

```
ics365fa2215@sp-cfsics:~/wk11$ pico stu.py
ics365fa2215@sp-cfsics:~/wk11$ python3 stu.py
Enter name: AAA
Enter first test score: 98
Enter second test score: 96
Enter name: BBB
Enter first test score: 94
Enter second test score: 99
Student name is:  AAA
First test score is:  96
Second test score is:  94
Grade = A
Student name is:  BBB
First test score is:  96
Second test score is:  94
Grade = A
```

References:
"Python Programming" by Richard L. Halterman, 2019
https://cs.appstate.edu/~rmp/cs2435/pythonbook.pdf

"Python Tutorial" by Guido van Rossum, 2012
http://marvin.cs.uidaho.edu/Teaching/CS515/pythonTutorial.pdf