***CSC 3020***

***Java Programming***

**Assignment 01**

**40 points**

**Due 09/18/2023 (11:45 A.M.)**

Assignment Objectives:

■■ To understand the meaning of Java language specification, API, JDK™, JRE™.

■■ To create, compile, and run Java programs.

■■ To explain the runtime errors.

■■ To obtain input from the console using the **Scanner** class.

■■ To explore Java numeric primitive data types: **byte**, **short**, **int**, **long**, **float**, and **double**.

■■ To distinguish between postincrement and preincrement and between postdecrement and predecrement.

■■ To cast the value of one type to another type.

■■ To declare **boolean** variables and write Boolean expressions using relational operators.

■■ To generate random numbers using the **Math.random()** method.

■■ To program using selection statements for a variety of examples.

■■ To combine conditions using logical operators (**!**, **&&**, **||**, and **^**).

■■ To implement selection control using **switch** statements.

■■ To examine the rules governing operator precedence and associativity.

**Solution to this assignment will not be posted on Canvas; however, any question can be discussed in the class upon request of a student.**

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

1. For non-programming questions, use a word file to type your answers. Don’t use the text box on the Canvas to answer the questions or to write comments, we will not read it.
2. State your answer clearly.
3. For programming questions, include only the source file for each problem.
4. Submit your file to the Canvas. You must submit your assignment on time; otherwise, you will receive zero. In addition, you cannot submit your file more than one time.
5. There will be several folders on the Canvas. You need to upload your file(s) using the correct folder on the Canvas.
6. Name each file: “Assignment Number(Question number(s))”.
7. To upload your file(s):

* In Course Navigation, click the Assignments link.
* Click the title of the assignment.
* Click the **Submit** Assignment button.
* Add **File**. ...
* Add Another **File**. ...
* **Submit** Assignment. ...
* View **Submission**.

**It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.**

**Make sure you review the Cheating & Plagiarism policy on Canvas.**

Answer question 1 on a text file (include question number and the answer only); write a program for each of Q.2 - Q.5. Convert .java files to text files. Submit 5 text files to Canvas by the due date.

1. **(12 points) Give a brief answer for each of the following questions:**
2. What is the difference between an interpreted language and a compiled language?
3. What does JDK stand for? What does JRE stand for?
4. What is the Java source filename extension, and what is the Java bytecode filename extension?
5. What is the command to compile a Java program?
6. What is the command to run a Java program?
7. What is the JVM?
8. If your program needs to read integers, but the user entered strings, an error would occur when running this program. What kind of error is this?
9. Show the output of the following code:

**double** amount = 5;

System.out.println(amount / 2);

System.out.println(5 / 2);

1. Can the following conversions involving casting be allowed?

**boolean** b = **true**;

i = (**int**)b;

**int** i = 1;

**boolean** b = (**boolean**)i;

1. What data types are required for a switch variable? If the keyword break is not used after a case is processed, what is the next statement to be executed?
2. What is y after the following switch statement is executed? Rewrite the code using an if-else statement.

x = 3; y = 3;

**switch** (x + 3) {

**case** 6: y = 1;

default: y += 1;

}

1. Why does the Math class not need to be imported?

**Programming Questions**

1. **(6 points)** The U.S. Census Bureau projects population based on the

following assumptions:

1. One birth every 7 seconds
2. One death every 13 seconds
3. One new immigrant every 45 seconds

Write a program to display the population for each of the next five years. Assume that the current population is 312,032,486, and one year has 365 days.

Here is a sample run:

year 1: 314812582

year 2: 317592679

year 3: 320372776

year 4: 323152872

year 5: 325932969

1. **(8 points)** Write a program that prompts the user to enter the seconds (e.g., 1 billion), and displays the number of years, days, hours, minutes, and seconds. For simplicity, assume a year has 365 days.

Here is a sample run:

Enter number of seconds (e.g., 1 billion) 1987000000

In 1987000000 there are: 63 years

In 1987000000 there are: 2 days

In 1987000000 there are: 16 hours

In 1987000000 there are: 26 minutes

In 1987000000 there are: 40 seconds

1. **(8 points)** Write a program that generates a lottery of a three-digit number. The program prompts the user to enter a three-digit number and determines whether the user wins according to the following rules:
2. If the user input matches the lottery number in the exact order, the award is $10,000.
3. If all digits in the user input match all digits in the lottery number, the award is $3,000.
4. If one digit in the user input matches a digit in the lottery number, the award is $1,000.

Here are sample runs:

Enter your lottery pick (three digits): 591

Lottery is 248

Sorry, no match

Enter your lottery pick (three digits): 987

Lottery is 637

Match one digit: you win $1,000

Enter your lottery pick (three digits): 683

Lottery is 836

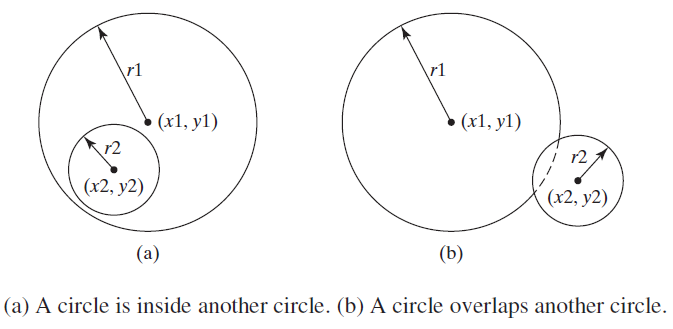
Match all digits: you win $3,000

Enter your lottery pick (three digits): 494

Lottery is 494

Exact match: you win $10,000

1. **(6 points)** Write a program that prompts the user to enter the center coordinates and radii of two circles and determines whether the second circle is inside the first or overlaps with the first, as shown in the below figure.

**

*Hint*:

circle2 is inside circle1 if the distance between the two centers < = **r1 -** **r2**

circle2 overlaps circle1 if the distance between the two centers <= **r1 + r2**.

Test your program to cover all cases.

Here are sample runs:

Enter circle1’s center x-, y-coordinates, and radius: 0.5 5.1 13

Enter circle2’s center x-, y-coordinates, and radius: 1 1.7 4.5

circle2 is inside circle1

Enter circle1’s center x-, y-coordinates, and radius: 3.4 5.7 5.5

Enter circle2’s center x-, y-coordinates, and radius: 6.7 3.5 3

circle2 overlaps circle1

Enter circle1’s center x-, y-coordinates, and radius: 3.4 5.5 1

Enter circle2’s center x-, y-coordinates, and radius: 5.5 7.2 1

circle2 does not overlap circle1