CS172 Final Exam
Summer 2022

Note: *This is an open book exam that must be finished independently. It is ok to use Internet or your previous quiz/program as the reference, but you should not ask help from other people. Submit the exam file and all Java program files to Canvas.*

Name Gunnar Yonker Score _____

1. (10 points) Single choices.

(1) Which of the following is an attribute declaration statement?
a. private temp;
b. public static void main(String args[])
c. import java.util.Scanner;
d. private double total;

(2) Which of the following is the constructor of class HardQuestion:
a. public Constructor (String question) { ….}
b. public HardQuestion (String question) {….}
c. public static void HardQuestion (String question) {….}
d. void HardQuestion (String question) {….}

(3) Suppose that you have declared an object house1 as an instance of the House class that has an attribute named as price, which of the following is correct in using the getter method:
a. house1.getPrice;
b. house1.getPrice(double price);
c. house1.getPrice();
d. getPrice(house1);

(4) Suppose that you want to write a program that calculates the product of two numbers. Fill in the blank:

```
public double getProduct(double x, double y){
    _____
}
```

a. return x;
b. return x*y;
c. return getProduct;
d. return y;

(5) Which of the following is a valid variable declaration and initialization?
a. int while=3;
b. int aCase3 = 4.2;
c. String sheepName = "Doley";
d. int 3$Var =5.0;

2. (5 Points) Write the results of the following statements (pay attention to integer division – e.g, 3*4/5 = 2, 3/5 = 0, 3.0/5 = 0.6):

(1) 8 + 5*3/(2*2)*3 = 17

(2) 4*5/ Math.pow(2.0, 3) *2 = 5

(3) 2*(int)(4.6) = 8

(4) 7/2*6/2 = 9

(5) 27 % 12 = 3

3. (30 points) Identify grammar/logic errors in the following codes and correct them. You should describe the errors first, and then show the correct code.

(1)

```
double gpa[] = {2.2, 3.8, 4.0, 3.6, 1.7, 2.8};
int countSuper = 0; // number of students whose GPA >= 3.5
for (int count = 0; count < 6;  count++){
     if(gpa[count] >= 3.5){
          System.out.print("\n You are super!");
     }  countSuper++;
          countSuper++;
}
System.out.print("\n Number of super students is: "+
countSuper);
```

countSuper++ is outside of the if statement loop so it adds 1 to that variable for each gpa no matter the value. Prints out 6 super students instead of 3.

Move countSuper++; under the System.out.print in the if loop.

(2)

```
public class TstArray {
     public static void main(String args[]){
          int a[] = {5, 4, 9};
               int avg;

          int total=0;
          for(int i=0; i< a.length; i++){
               total = 0;
               total += a[i];
          }

          avg = total/a.length;
          System.out.println("Average value is " + avg);
     }
}
```

int total; is not initialized, easy fix to initialize by changing it to int total = 0;

Nothing will print out so should add print line for avg, but will show avg is 3 due to setting the total to 0 in the for loop.

To fix this, total = 0; can be removed now that it is initialized in the declaration.

New and correct avg is 6.

(3)

```
int value = 0;
do{
      value += 10;
}while(value > 5);

System.out.println(value);
```

to fix logic error

```
int value = 0;
do {
    value += 10;
  System.out.println(value);
} while (value <100);
```

10
20
30
40
50
60
70
80
90
100

In this do while loop the do sets the value at 10 which is greater than 5 in the while loop, this is a logic error because the while loop condition will cause the loop to never stop. Depending on the objective of the code it could be changed, if you move the system.out.println below the value +=10; and change while(value <100) it would print this then stop

(4) The following code was intended for inputting scores for 20 students. If an invalid value was input, the program should keep reminding the user to re-input till a valid value is input.

```
//Create Scanner object
     Scanner myScanner = new Scanner(System.in);

     //Declare variables
     final int studentNumber = 20;
     int scores[] = new int[studentNumber];

     //Use while loop to allow user keep trying till valid input
     while(true)
     {
          for(int i=0; i<scores.length; i++) {
               System.out.println("Score for student "+(i+1));
               scores[i] = myScanner.nextInt();

               if( 100 >= scores[i]> 0){
                    break;
               }
               //Otherwise continue the loop
          } // end for
     }//end while
```

if (scores[i] >= 0 && scores[i] <= 100){

The if statement is formatted incorrectly, should use && so score entry is between 0 and 100 to break. Currently with the break the code will indefinitely put in a score for student 1 because it breaks out of the for loop not the while, which makes it continuously repeat (an more while loop, example correction below

```
Scanner myScanner = new Scanner(System.in);
final int studentNumber = 20;
int scores[] = new int[studentNumber]
for(int i=0; i<scores.length; i++) {
while(true){
  System.out.println("Score for Student " + (i+1));
  scores[i] = myScanner.nextInt();
if(scores[i] >=0 && scores[i] <=100){
  break;
}
else if(scores[i] <0 || scores[i] >100){ //makes sure score is positive and below 100
  System.out.println("Invalid Score, try again");
}
}
}
```

(5) The following code was intended for obtaining the number of times you have to divide 1 by 2 to get a value less than one ten-thousands (0.0001). Please *point out errors* and *correct/complete* the code.

```
//Declare variables
int number = 1;
double quotient = number/2;
while(quotient>=0.0001);
{
    quotient = quotient/2;
}
```

no need since while does this

semi-colon ends while

When you run this code it will print out 0.0 as the answer. To complete the code you will also need to implement a counter to get the answer of how many divisions. this prevents while from running

Correct Code

```
int count = 0;
double quotient = 1;
while (quotient >= 0.0001){
    quotient = quotient /2;
    counter ++;
}
System.out.println("You divide 1 by 2, " + counter + " times to get a "1" value less than 0.0001)
```

(6) The following is someone's definition (with quite some errors) for the Checking class.

```
public class Checking {

    //Only one attribute (for simplicity, nothing wrong here)
    private double balance;

    public void Checking(double iniBalance){
        setBalance(iniBalance);

    }
   // Define getter and setter
    public double getBalance(){
        return balance;
    }

    public void setBalance(double newbalance){
        double balance = 0;

        if(newbalance>0){      //validity check
        this.balance = newbalance;
        }
        else{
            System.out.println("Your input is invalid!");
        }    this.balance = 0;
    }

}
```

void should not be here for the constructor.

the return value type is double for balance not string which is seen in the attribute declaration for balance

not needed because balance already exists

makes sure balance is above 0 then sets it

add in to set balance to 0 if invalid input

* looks cleaner to change "set balance" to "setBalance" and "get Balance"

(7) Following is ill-written code intended for judging a number is prime or not. Note that you should point out the problems first and then correct the code.

```
//define variables
int number;

//Create scanner
Scanner myScanner = new Scanner(System.in);

System.out.println("Please input a positive integer");
number = myScanner.nextInt();
if(number<0){
    System.out.println("Number must be positive"); //make sure input is positive, if not remind user
    return;
}
for(int counter = 2; counter < number; counter++)
{
    if (number%counter == 0 ){
        System.out.println(number+" is NOT prime!");
    }
    else{
        System.out.println(number+" is prime!");
        break;
    }
}//end for
//correct code after if statement
{System.out.println(Number + " is prime!");
    return;
}
}
```

can simplify by dividing number by 2 in for loop counter should start at 2

(number/2)

this prime check needs to move out of the for loop and remove else statement, looks like this

break should now change to return;

4. (15 point(s)) Read the following Java programs and write out the printout results.

(1)
```
int val = 2;
switch( val ) {
  case 1: System.out.print("Today ");
  case 2: System.out.print("Is ");
  case 3: System.out.print("Really ");
  default: System.out.print("Hot ");
          break;
}
```

Printout Result

Is Really Hot

(2)
```
int number[] = {2,3,6,7,9,10};

for (int i=0; i< number.length; i++) {
    if (i%2 != 0) {
        System.out.print(number[i]+" ");
    }
}
```

Printout Result

3 7 10

(3)

```
int value = 0;
int i;
for(i=0; i<10; i+=4){
    value = value + i;
}

System.out.println("Final value = " +value);
```

Print out Result

Final value = 12

(4)

```
public class MinMax{
        private double x,y;

        public MinMax( double a, double b ) {
                this.x = a ;
                this.y = b ;
        }
        public double min() {
                double minimum = 0 ;
                if(x >y)
                        minimum = y ;
                else
                        minimum = x ;
                return minimum;
        }
        public double max() {
                double maximum = 0 ;
                if(x <y)
                        maximum = y ;
                else
                        maximum = x ;
                --
                return maximum;
        }
}
```

```
public class Demo{
    public static void main(String args[]){
        double num1, num2;
        num1 = 21.5 ; num2 = -24;

        MinMax obj1 = new MinMax(num2, num1) ;

        System.out.println("Minimum is" + obj1.min() ) ;
        System.out.println("Maximum is" + obj1.max() ) ;

        double minimum= 0 ;
        double maximum = 0 ;
        obj1.min() ;
        obj1.max() ;

        System.out.println("Minimum is" + minimum) ;
        System.out.println("Maximum is" + maximum) ;

    }
}
```

Printout Result

Minimum is-24.0
Maximum is21.5
Minimum is 0.0
Maximum is 0.0

(5)

```
int value = 10, j = 3;
while(value <=20){
        value += j;
        j++;
}
System.out.println("Final value = " +value);
```

Printout Result

Final Value = 22

Programming on the Computer (40 points)

Guidelines:

(1) You must finish the program independently. It is ok to use Internet or your previous program as the reference, but you should not ask help from other people.

(2) At the head part of your java file, add brief comments that include author name, creation date, the main purpose of this program.

(3) Compile and debug your files before uploading your .java files.

5. Create a class and a main() method that implements the following (define the local variables or additional arrays as you see necessary

(1) Create a salary array (integer type) with 8 elements.

(2) Ask the user to input value from keyboard for each array element. If the input value is not between 0 and 15,000, keep asking the user to re-input till a valid value is obtained (hint: use a while loop).

(3) If one's salary is higher than $8,000, he/she get 8% of his/her salary for bonus. Otherwise the bonus rate is 12%.

(4) After the salary information has been input for everyone, print out the salary and bonus for everyone. Each record in a separate line.

(5) Find out the highest bonus value and print it out.

File : SalaryMain.java

6. Age calculation. You must use the object oriented programming for this problem.

(1) Create a class called Date that

a. Includes three pieces of information as data members-a month (type int), a day (type int) and a year (type int).

b. Have a constructor with three parameters to initialize the three data members. Call setters inside to initialize each attribute.

c. Provide a *set* and a *get* method for each data member. Assume that the values provided for the year and day are correct, but ensure that the month value is in the range 1-12; if it isn't, set the month to 1.

d. Provide a member function displayDate that displays the month, day and year separated by forward slashes (/).

e. Provide a member function called calculateAge with three parameters that indicate a future date (represented as month, date, and year). Calculate one's age based on one's birth date and this future date. Inside the method simply print out one's age in the format of how many years and how many days. Note that here you are not required to judge a year is a leap year or not. Simply count February as 28 days for every year.

(2) Create a AgeCalculation class with a main method, inside which:

a. Ask user to input month, date, and year of one's birth date, and then create a Date object (say birthday) using the input values. Call the object's displayDate method to display the date information in the formatted way.
b. Ask one to input a future date (month, day, and year), use the calculateAge method of the Date class to check one's age (how many years and how many days).

Files: Date.java
AgeCalculation.java