| Points off | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Total off | Net Score |
|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |  |

# CS 305j – Midterm 1 – Spring 2010

Your Name________________________________

Your UTEID ________________________________

Instructions:

1. Please turn off your cell phones
2. There are 7 questions on this test.
3. You have 60 minutes to complete the test.
4. You may not use a calculator.
5. Please make your answers legible.
6. When code is required, write Java code.
7. The proctors will not answer questions.

1. Expressions. 2 points each, 20 points total. For each Java expression in the left hand column, indicate its value in the right hand column. **<u>Be sure to show a constant of the appropriate type. For example, 7.0 rather than 7 for a double and "7" instead of 7 for a String.</u>**

A. `4 + 2 * 3` ______________________________

B. `10 + 4 / 10` ______________________________

C. `20 / 7` ______________________________

D. `200 / 400 / 10 / 2 / 100` ______________________________

E. `20 % 7 + 7 % 20` ______________________________

F. `"UTCS" + (3 * 7)` ______________________________

G. `1 + 2 + "UT" + 2 + 1` ______________________________

H. `200 % (10 * 10)` ______________________________

I. `1.7 + 2 * 2 / 4` ______________________________

J. `5.0 / 2` ______________________________

2. Parameters Simulation. (18 points) Consider the following program:

```
public class ParameterQuestion{

      public static void main(String[] args){
            int x = 3;
            int y = 1;
            System.out.println(alpha(x, y));

            System.out.println(x);

            System.out.println(y);

            System.out.println( beta(2, 6) );

            x = 4;
            y = 2;
            System.out.println( beta(x, y) );

            int z = gamma(x);
            System.out.println(z);

            x = 3;
            y = 4;
            delta(x,y);
      }

      public void int alpha(int a, int b){
            a = a + 2;
            int z = a + b;
            System.out.println("A: " + z);
      }

      public static int beta(int w, int z){
            int x = w * z;
            return x;
      }

      public static int gamma(int x){
            int y = x * 3;
            x += 3;
            return Math.max(y, x);
      }

      public static void delta(int x, int y){
            int w = alpha(x, x);
            int z = gamma(y);
            int result = w + z;
            System.out.println("D: " + result);
      }
}
```

List output of program here:

In the box to the right list what output is produced when the program is run.
List the output in the order it would appear on the screen.

3. Loops Simulation. 12 points. Consider the following method:

```
public static void loop(int x, int y){
     for(int i = 1; i <= x; i++)
     {    for(int j = 1; j <= i; j++)
          {    System.out.print("+");
          }

          for(int j = 1; j <= y; j++)
          {    System.out.print(i);
          }

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

What is printed out by the following method calls?

Method Call                                        Output to screen

`loop(0, 5);`

-------------------------------------------------------------------

`loop(2, 2);`

-------------------------------------------------------------------

`loop(4, 5);`

-------------------------------------------------------------------

`loop(3, -1);`

4. Debugging. (12 points)

A. (2 points) The following method causes a runtime error. Briefly explain why the run time occurs.

```
public static void show(int x)
{    int div = 20 % 5;
      int res = x / div;
      System.out.println(res);
}
```

B. (2 points) The intent of the following method is to print out numbers in a pyramid pattern such as this (assuming max is 3)

```
1
22
333
```

Instead the method goes into what appears to be an infinite loop, printing out 1's on a single line. Briefly explain the logic error that causes this behavior.

```
public static void nums(int max)
{   for(int i = 1; i < max; i++)
    {   for(int j = 1; j < i; i++)
        {   System.out.print(j);
        }
        System.out.println();
    }
}
```

C. (8 points) The following program contains 5 syntax errors. Circle the syntax errors and briefly state what the syntax error is.

```
public static void main(String[] args)
{
    string name = "Isabelle";

    int 2010Age = 12;

    int quarter = 10.5 / 4.0;

    int x;

    int y = x * 3;

    four(int i = 0; i < 12; i++)
        System.out.print(i + ", ");
}
```

5. Programming. (10 points) Write a method that calculates and returns the force of attraction between two particles based on Newton's Law of Universal Gravitation.

According to Newtoon's Law of Universal Gravitation the force of attraction between two particles is equal to:

$$F = G\frac{m_1 m_2}{r^2}$$

Where F is the force of attraction between the two particles, G is the gravitational constant, $m_l$ is the mass of the first particle, $m_2$ is the mass of the second particle, and r is the distance between the centers of the two particles.

G is equal to 6.67 x $10^{-11}$. You can assign doubles in Java values in scientific notation. For example, Planck's constant is 6.67 x $10^{-34}$. A Java double could be assigned this value as follows:

```
double p = 6.67E-34;
```

Write a method that given the mass of the two particles and the distance between them, returns the force of attraction between the two particles based on Newton's Law of Universal Gravitation. The masses and distance will all be given as `doubles`. (Assume the units for all values are correct.)

6. Programming (20 points) Write a method named `getLastHalfRev` that given a `String`, returns a new `String` that consists of the second half of the given `String` with the characters in reverse order. If the `String` has an odd number of characters the middle character is included in the returned String.

Recall the characters in a String use zero based indexing. Thus in the `String "Cat"` the character 'C' is at index 0, the character 'a' is at index 1, and the character 't' is at index 2.

The only methods you need from the String class are the `length` method and the `charAt` method.

`int` **`length`**`()` Returns the length of this string.
`char` **`charAt`**`(int index)` Returns the `char` value at the specified index.

For example:

```
String p = "Cat";
System.out.println(p.length());
```

Here are some example of method calls to `getLastHalfRev`:

```
getLastHalfRev("Cat"); // returns "ta"
getLastHalfRev("Cats"); // returns "st"
getLastHalfRev("Cats12"); // returns "21s"
getLastHalfRev("Cats123"); // returns "321s"
getLastHalfRev("A"); // returns "A"
getLastHalfRev(""); // returns "", an empty String
getLastHalfRev("12_Secrets_124"); // returns "421_ste"
```

7. Guest Speaker. (8 points). We had a guest speaker in class, Professor Peter Stone of the UT Computer Science Department. He described the research of one of his graduate students, Kurt Dresner, and the problem of Autonomous Intersection Management.

1. Circle any of the following statements that are **not true** in regards to the Autonomous Intersection Management project.

A. The Federal Highway Administration has shown interest in the project and has provided some funding for it.

B. The simulation assumes the Intersection Management program will have complete control all autonomous vehicles in the range of the intersection.

C. The research did not consider how to handle situations involving some vehicles being driven by computer programs and some vehicle being driven by humans.

D. Due to changes in requirements, such as the ability to have a simulation with multiple intersections, the simulation program has been rewritten many times.

E. The primary factor used to compare different intersection protocols was the average delay time per car when using that protocol.

2. Circle any of the following statements that are **true** in regards to Artificial Intelligence and the Autonomous Intersection Management project.

A. Machine learning describes a software program that can change and adapt as it gets results from attempts to solve some problem.

B. The Autonomous Intersection Management research did not consider how to handle emergency vehicles such as police, fire trucks, or ambulances.

C. The Autonomous Intersection Management research did not consider how to report current traffic conditions to a city wide central database and how that central database could more effectively route traffic throughout the city.

D. In the field of artificial intelligence an autonomous agent is a generally thought of as a computer program that can sense its environment, makes, decisions, and takes actions based on those decisions to achieve some goal.

E. The Autonomous Intersection Management research did not consider how to handle adverse weather conditions such as rain, ice, or snow.

Scratch Paper

Scratch Paper