

Panel 1

CSAS 1111 - Review

- ✓ Programming Language, Source Code
- ✓ Compiling and Executing a program
- ✓ Basic Data Types, Strings Binary #'s
- ✓ Assignment and Arithmetic
- ✓ Conditional Execution, Loops
- ✓ Input and Output
- ✓ Methods and Parameters
- ✓ Fields and Scope
- ✓ Arrays
- ✓ Program Design

1

Panel 2

Basics

1. Describe the complete process of creating and executing a working program. In other words, how is the source code file created, what names must be used, how is the class file created, and how is the class file executed.

Name.java contains source code in Java

Compile : Name.java → Name.class

Execute : Name.class

"java Name" from C:\ (DOS Box)

2

Panel 3

1. Convert the numbers 321 and 33 to their binary representations, and the numbers $(010010001)_2$ and $(11011011)_2$ to their usual representation.

128	64	32	16	8	4	2	1
		1	0	0	0	0	1

1 1 0 1 1 0 1 1 = 128 + 64 + 16 + 8 + 2 + 1

3

Panel 4

1. What is the value of z at the end of this code fragment?

and result

```
int z = 23;
int x = 7, y = 53;
```

```
x *= y;
y /= 2;
x++;
z += (x + y);
```

```
x = 5;
y = 9;
z = 2;
```

```
boolean r = false, result;
result = !( ( f || (y < x) ) && ( 5 >= x ) ); true
result = ( ( x + y ) >= ( 16 - z ) );
```

$x = \cancel{7} \times \cancel{53} = 2 \times 5 = 10$ $x + y = 10 + 53 = 63$

$y = \cancel{53} / 2 = 9$

$z = \cancel{2} + 10 + 9 = 21$

4

Panel 5

1. Write a program that has three `double` variables containing some values and prints to the screen the mathematical *sine* of the first one, the *cosine* of the second one, and the *tangent* of the third number. Also display the sum of the squares of *sine* and *cosine* of the three double numbers.

```

=> public class Bert
{
    => public static void main (String args[])
    {
        double x = 1.0, y = 2.0, z = 3.0;
        System.out.println (Math.sin(x)) ;
    }
}

```

because said "program"

5

Panel 6

- a) what is the value of x, y, and z if x = 4, y = 9, z = false.

```

int x, y; ✓
boolean z; ✓

```

```

if (x > 3)
{
    z = true;
    x = 7;
    y *= 4;
}
else if (x == 4)
{
    z = false;
    x = 9;
}
else
{
    z = true;
    y = 8;
}

```

x=4
y=9
z=false

xx = ~~32~~ ~~16~~ ~~8~~ ~~4~~ ~~2~~ 1

```

double xx = 32;
while (xx > 1.0) ✓
    xx /= 2;
System.out.println("xx = " + xx);

```

6

Panel 7

1. Write a ~~program that~~ ^{some code that} displays all numbers divisible by 4 between 1 and 100 using a for loop.

```
for (int i = 1; i <= 100; i++)
{
    if ( (i % 4) == 0 )
        System.out.println(i);
}
```

or

```
for (int i = 4; i <= 100; i += 4)
{
    System.out.println(i);
}
```

7

Panel 8

1. Describe the difference (for strings) between "==" and ".equals" and "="

== equality, = assignment

① .equals checks if chars are same in same position

s.equals(+)

1. Create a ~~program~~ ^{code} that displays a ^{given} string ^s in reverse order, one character per line, to the screen.

```
for (int i = s.length() - 1; i >= 0; i--)
    System.out.println(s.charAt(i));
```

8

Panel 9

1. Describe, in your own words:

- a) why methods are useful.
- b. how to define a method
- c. the difference between the body of a class, the body of a method, and the header of a method.

```

Public class X
{
  public static RehanType name (input list)
  {
    blah
  }
}
  
```

body class (pointing to the class braces)

header of method (pointing to the method signature)

body method (pointing to the method body)

9

Panel 10

```

public static void methodA(int i)
{ // stuff }
  
```

```

public static double methodB(char c, double x)
{ // stuff }
  
```

Give an example how you would call each method.

```

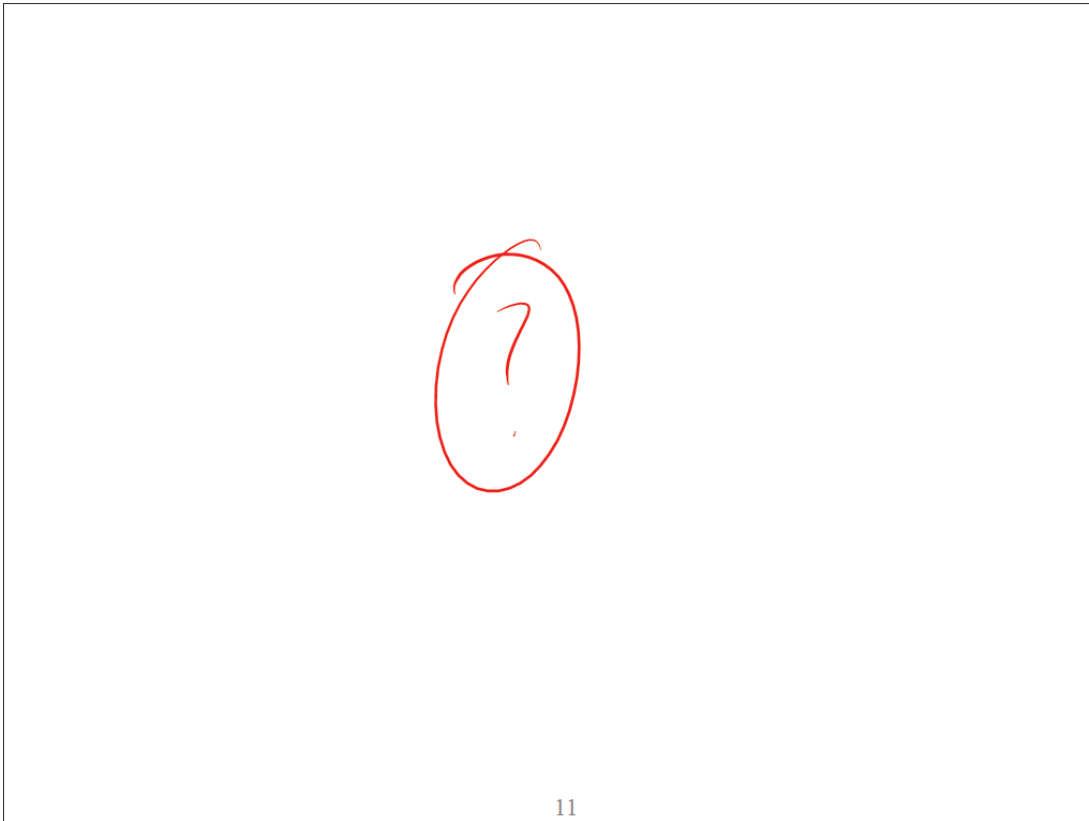
methodA(1);
  
```

```

double x = methodB('c', 2.7);
  
```

10

Panel 11



Panel 12

1. Describe, in your own words:

- a. the difference between a field variable and a local variable
- b. when to use fields and when to use local variables
- c. why local variables are important
- d. why fields are important
- e. the scope of a field
- f. the scope of a local variable

f. from its declaration to
end of block it's declared in.

```
Ⓟ for (int i = 1; i < 10; i++)  
  { double z = i / 2.0;  
    }  
  → System.out.println(z); X  
  → System.out.println(i); X
```

Panel 13

```

public class FlavoredScope
{
    public static int x = 4;
    public static int y = 8;
    public static int z = 2;

    public static void mints(int x, int z, int y)
    {
        winterfresh(y,z);
        x = peppermint(y);
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
    }

    public static void winterfresh(int x, int y)
    {
        int z = -1;
        x = y + z;
    }

    public static int peppermint(int y)
    {
        y *= z;
        return y;
    }

    public static int cherry(int y, int z, int x)
    {
        x = y;
        y = x;
        z *= y;
        return z;
    }

    public static void main( String args[ ] )
    {
        mints(z, y, cherry(x, y, z));
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
    }
}

```

13

How many
a) fields
b) methods
Output!

Killer!
at home!

Panel 14

1. What is the exact syntax:
 - a. to declare an array of five integers
 - b. to declare an array without initializing it
 - c. to declare an array containing the following three strings: "Java" "Is" "Fun"
 - d. to specify the size of an array that was not initialized
 - e. to get the value from an array element
 - f. to create a method with an array as input type
 - g. to call a method that uses an array as an input type
 - h. to find the length of an array

HW

14

Panel 15

1. If an array is of size 25, what are the index numbers to access the elements of the array?

- a) 0-24
- b) 1-25
- c) 2-26
- d) 3-27
- e) 4-28

1. Consider the following code segment:

```
int A[] = new int[10];
A[0] = 1;
A[1] = 2;
for (int i = 2; i < A.length; i++)
{   if ((i % 2) == 1)
    A[i] = A[i-1] + A[i-2];
    else
    A[i] = 2 * A[i-1];
}
```

Recall that $(i \% 2)$ returns 0 if i is even and 1 if i is odd. What is the value of $A[4]$ after the above code segment executes? How about $A[9]$? What about $A[10]$?

HLW