**AP Computer Science – Methods “Big Ideas”**

1. Each method has three main components: A return type, a method name and a parameter list.
2. We only send values, not variables into methods.
3. Void methods do not have a return statement.
4. You can call one method from within another method.
5. If your method has a return type, make sure it ALWAYS returns a value.
6. We can overload a method (create several different methods with the same name) as long as each method has a different parameter list.
7. When a method is called, we have to understand what is being sent into the method and what is coming back.
8. A method that calls itself is called a *recursive* method.

Problem # 1 - Big Ideas: 1, 2, 7

**public** **static** **void** main(String[] args) {

 **int** x = 3;

 *squareIt*(x);

 System.***out***.println(x);

 }

**public** **static** **int** squareIt(**int** a){

 **return** a\*a;

 }

What is the output of the above code? Explain.

Problem # 2- Big Ideas: 1, 5

Identify the error in this method.

**public** **static** **boolean** gameOver(**int** playerScore, **int** computerScore)

{

**if**(playerScore==10 || computerScore==10) **return** **true**;

 **if**(playerScore < 10 && computerScore < 10) **return** **false**;

}

Problem # 3- Big Ideas: 1, 6

Can I put both of these methods in the same Java program? Explain.

**public** **static** **double** calculateTax(**double** sale)

{

 **return** sale \*1.06;

}

**public** **static** **int** calculateTax(**double** sale)

{

 **return** (**int**)(sale \* 106);

}

Problem # 4- Big Ideas: 1, 6

Can I put both of these methods in the same Java program? Explain.

**public** **void** greet(String name)

{

System.***out***.println("Hello " + name);

}

**public** **void** greet(String firstName, String lastName)

{

 System.out.println("Hello " + firstName + " " + lastName);

}

Problem # 5 – Big Ideas: 1, 6

What is the output of this program? Explain.

**public** **static** **void** main(String[] args)

{

 **int** num = 48;

 **while**(isEven(num))

 {

 cutInHalf(num):

 }

 System.out.println(num);

}

**public** **static** **boolean** isEven(**int** a)

{

 **return** a%2==0;

}

**public** **static** **int** cutInHalf(**int** a)

{

 **return** a/2;

}

Problem # 6 – Simplify this method. It can be done in one line of code!

 **public** **static** **boolean** hasIntegerSquareRoot(**double** num)

 {

 **double** sq = Math.*sqrt*(num);

 **int** sq2 = (**int**)(Math.*sqrt*(num));

 **if**(sq==(**double**)sq2) **return** **true**;

 **else** **return** **false**;

 }

**public** **static** **boolean** hasIntegerSquareRoot(**double** num)

 {

 }

Problem # 7 – Follow the logic to determine the output. Use substitution as values are sent in and returned. Make sure you follow the flow of code in the same order as the computer will. Enter the output into the box on the right.

**public** **class** FollowLogic

{

 **public** **static** **void** main(String[] args)

{

 **boolean** b = **false**;

 **int** x = *calculate*(b, *find*(11));

 System.***out***.println(x);

 }

**public** **static** **int** find(**int** a)

{

 **return** a/3;

}

**public** **static** **int** calculate(**boolean** a, **int** b)

{

 b = *adjust*(b);

 a = !a;

 **if**(a) **return** *change*(b);

 **else** **return** b -10;

}

**public** **static** **int** change(**int** a)

{

 **return** a + 7;

}

**public** **static** **int** adjust(**int** a)

{

 **if**(a%2==0) **return** a-1;

 **return** 2\*a;

}

}

Problem # 8

Calculate the output. Write down each method call to track the program’s progression. This one is hard!!!!

**public** **class** Recursion {

 **public** **static** **void** main(String[] args)

 {

 System.***out***.println(*doMath*(6));

 }

 **public** **static** **int** doMath(**int** a)

 {

 **if**(a==0) **return** 1;

 **else** **return** a \* *doMath*(a-1);

 }

}

Name:

**Methods – Check for Understanding**

For each of the “Big Ideas” below, write your confidence level from 1 (no confidence) to 10 (100% confidence).

|  |  |
| --- | --- |
| Big Idea | Confidence Level |
| Each method has three main components: A return type, a method name and a parameter list. |  |
| We only send values, not variables into methods. |  |
| Void methods do not have a return statement. |  |
| You can call one method from within another method. |  |
| If your method has a return type, make sure it ALWAYS returns a value. |  |
| We can *overload* a method (create several different methods with the same name) as long as each method has a different parameter list. |  |
| When a method is called, we have to understand what is being sent into the method and what is coming back. |  |

Name:

**Methods – Check for Understanding**

For each of the “Big Ideas” below, write your confidence level from 1 (no confidence) to 10 (100% confidence).

|  |  |
| --- | --- |
| Big Idea | Confidence Level |
| Each method has three main components: A return type, a method name and a parameter list. |  |
| We only send values, not variables into methods. |  |
| Void methods do not have a return statement. |  |
| You can call one method from within another method. |  |
| If your method has a return type, make sure it ALWAYS returns a value. |  |
| We can *overload* a method (create several different methods with the same name) as long as each method has a different parameter list. |  |
| When a method is called, we have to understand what is being sent into the method and what is coming back. |  |