Montgomery College, CMSC 203
Worksheet 1
Module 6

Objectives

- Methods
- Passing arguments to methods
- Returning a value from a method

Warm-up Question – (Dialog Box and random numbers)

1. Which of the following properly uses a DialogBox?

a) `JOptionPane.showMessageDialog(String,“Hello World");`
b) `JOptionPane.showMessageDialog(null, "Hello World");`
c) `JOptionPane.System.out.println(null, "Hello World");`
d) `JOptionPane.showMessageDialog(“Hello World”,null);`

Answer: B

2. Write a java statement that will randomly generate an integer between 5 and 10?

Answer:

```
Random r = new Random();
int num = r.nextInt(10) + 5;
```

Concept Questions

1) Methods are commonly used to:
A) speed up the compilation of a program
B) break a problem down into small manageable pieces
C) emphasize certain parts of the logic
D) document the program

Answer: B

2) Values that are sent into a method are called:
A) variables
B) arguments
C) literals
D) types
Answer: B

3) When an argument is passed to a method:
A) its value is copied into the method's parameter variable
B) its value may be changed within the called method
C) values may not be passed to methods
D) the method must not assign another value to the parameter that receives the argument

Answer: A

4) What is wrong with the following method call?

```
displayValue (double x);
```

A) There is nothing wrong with the statement.
B) `displayValue` will not accept a parameter.
C) Do not include the data type in the method call.
D) `x` should be a `String`.
Answer: C

5) Given the following method header, which of the method calls would be an error?

```
public void displayValues(int x, int y)
```

A) `displayValue(a,b); //` where `a` is a `short` and `b` is a byte
B) `displayValue(a,b); //` where `a` is an `int` and `b` is a byte
C) `displayValue(a,b); //` where `a` is a `short` and `b` is a long
D) They would all give an error.
Answer: C

6) Which of the following would be a valid method call for the following method?

```
public static void showProduct (int num1, double num2)
{
    int product;
    product = num1 * (int)num2;
    System.out.println("The product is " + product);
}
```

A) `showProduct(5.5, 4.0);`
B) `showProduct(10.0, 4);`
C) `showProduct(10, 4.5);`
D) `showProduct(33.0, 55.0);`
Answer: C

7) A special variable that holds a value being passed into a method is called what?
A) Modifier
B) Parameter
C) Alias
D) Argument
Answer: B

8) A parameter variable's scope is:
A) the method in which the parameter is declared
B) the class to which the method belongs
C) the `main` method

D) All of the above
Answer: A

9) The lifetime of a method's local variable is:
A) the duration of the program
B) the duration of the class to which the method belongs
C) the duration of the method that called the local variable's method
D) only while the method is executing
Answer: D
10) Local variables:
A) are hidden from other methods
B) may have the same name as local variables in other methods
C) lose the values stored in them between calls to the method in which the variable is declared
D) All of the above
Answer: D

11) Which of the following values can be passed to a method that has an `int` parameter variable?
A) `float`
B) `double`
C) `long`
D) All of the above
E) None of the above
Answer: E

12) What will be returned from the following method?

```
 public static double methodA()

{
  double a = 8.5 + 9.5;
  return a;
}
```

A) 18.0
B) 18 (as an integer)
C) 8
D) This is an error.
Answer: A

13) When a method tests an argument and returns a `true` or `false` value, it should return:
A) a zero for `true` and a one for `false`
B) a `boolean` value
C) a zero for `false` and a non-zero for `true`
D) a method should not be used for this type test
Answer: B

14) In the following code, `Integer.parseInt(str)`, is an example of:

```
int num;
string str = "555";
num = Integer.parseInt(str) + 5;
```

A) a value-returning method

B) a `void` method

C) a local variable

D) a complex method

Answer: A

15) Given the following method header, which of the method calls would be an error?

```
public void displayValues(double x, int y)
```

A) `displayValue(a,b); //` where `a` is a `long` and `b` is a `byte`

B) `displayValue(a,b); //` where `a` is an `int` and `b` is a `byte`

C) `displayValue(a,b); //` where `a` is a `short` and `b` is a `long`

D) They would all give an error.

Answer: C

16) Which of the following would be a valid method call for the following method?

```
public static void showProduct(double num1, int num2)
{
   double product;
   product = num1 * num2;
   System.out.println("The product is " +
                  product);
}
```

A) `showProduct("5", "40");`

B) `showProduct(10.0, 4.6);`

C) `showProduct(10, 4.5);`

D) `showProduct(3.3, 55);`

Answer: D

17) Java uses *pass by value* for passing parameters on a method call. What does this mean if the parameter is a primitive data type? What does this mean if the parameter is a reference to an object?

Answer: *Pass by value* means that the value associated with the actual parameter is copied into the memory location of the formal parameter when the method is called. The formal and actual parameters occupy distinct locations in memory. If the parameter is a primitive data type, any modifications made to the formal parameter during the execution of the method will most likely not be reflected in the actual parameter. If the parameter is a reference to an object, then it holds the location of an object. While the formal and actual parameters will be distinct, they will both hold the address of the object and will be aliases. If the method modifies the object, then those modifications will be apparent when referencing the object using the actual parameter after the method terminates.

18) What is the output of the following code?

```
public class PassByValue {

      public static void main(String[] args) {
          int x = 1;
          int y = 2;

          modifyPrimitiveTypes(x,y);

          System.out.println(" x = " + x + " ; y = " + y );
      }
      private static void modifyPrimitiveTypes(int x, int y)
      {
          x = 5;
          y = 10;
      }
}
```

Answer:

x = 1 ; y = 2

19) Following code represent a class `Number`: What is the output of the following code?

```
public class Number {
      private int value;

      public int getValue() {
            return value;
      }

      public void increaseValue(int n) {
            value += n;
      }
}
```

What is the output of the following code?

```
public class NumberDriver {

      public static void main(String[] args) {
             Number num1 = new Number();
             System.out.println(num1.getValue());
             passObject(num1, 5);
             System.out.println(num1.getValue());

      }
      public static void passObject(Number numObject, int n)
      {
            numObject.increaseValue(n);
```

```
    }

}
```

Answer:
0
5

Programming Questions

1) Write the definition of a method `printGrade`, which has a `char` parameter and returns nothing. The method prints on a line by itself the message string `Grade:` followed by the `char` parameter (printed as a character) to standard output. Don't forget to put a new line character at the end of your line.

Answer:

```
void printGrade(char c) {
            System.out.print("Grade: "+c);
            System.out.println();
        }
```

2) Given that a method receives three parameters `a, b, c,` of type `double`, write some code, to be included as part of the method, that determines whether the value of "b squared" – 4ac is negative. If negative, the code prints out the message "`no real solutions`" and returns from the method.

Answer:

```
if((Math.pow(b,2) - 4 * a * c) < 0)
{
    System.out.print("no real solutions");
}
```

3) Write a method `min` that has two `String` parameters and returns the smaller. Hint use the `compareTo` method of the `String` Class.

Answer:

```
public String min(String a, String b)
{
     if (a.compareTo(b)<0  )
     {
          return a;
     }

     else
     {
          return b;
     }
}
```

4) Write a method called `isPalindrome` that accepts a `String` as a parameter and returns `true` if the `String` is a palindrome, and `false` otherwise. You may assume that the entered String consists entirely of lowercase letters (meaning it contains no numbers, spaces, punctuation, etc). Hint: write code that creates a new string that is the original string reversed, and then check to see if the two strings are equal. Create a pseudo Code first for the body of the method:

Answer:

```
public boolean isPalindrome(String s)  {
     String reverseS = new String("");

     for(int i = s.length()-1; i >= 0; i--)
          reverseS += s.charAt(i);

     return reverseS.equals(s);
}
```