

ExploreMethods.pdf

Method 1

```
public static double getDouble(Scanner myScan) {  
    //we want to get a deposit and withdrawal amount, and will keep  
    asking for it, basically do the same as getInteger just replace it with  
    double  
    while(!myScan.hasNextDouble()) {  
        System.out.println("Invalid Input, please enter a double value  
integer");  
        myScan.next(); // again to clear invalid  
    }  
    return myScan.nextDouble(); // again, to return the double value  
}
```

Here, I used public static double getDouble(Scanner myScan)

This is a method named getDouble where Scanner myScan is the parameter (asking for a scanner input), and having double means it expects a double value.

Modifiers : public

Return type : double

Method Name : getDouble

Param List : Scanner myScan

Signature : getDouble(Scanner myScan)

```
public static int getInteger(Scanner myScan) {  
    while(!myScan.hasNextInt()) {
```

Modifiers: public

Return type: int

Method name: getInteger

Param list: Scanner myScan

Signature : getInteger(Scanner myScan)

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

Modifiers : public

Return type : void

Method name : main

Param list : String [] args

Signature: main(String [] args)

```
public static String getStringName(Scanner myScan) {\
```

Modifiers : public

Return type : String

Method name : getStringName

Param list : Scanner myScan

Signature: getStringName(Scanner myScan)

1. The benefits of using a method are correctness (less chance to miss some character or semicolon if you just call it instead of typing it again) and saving time. Also, it looks better
2. Defining a method is done within the same class but outside the main method. The syntax is like so, and calling it is like so.

```
a.     access_modifier return_type method_name(parameters) {  
b.         // code block  
c.     }  
  
return_type result = method_name(arguments);
```

3. Return type of main method is void, because it doesn't need to return a single value, but a bunch of them
4. Not writing a return type in a value returning method is that the compiler will have an error. If it's not specified then it can't be called in places where it's supposed to expect a specific return
5. Yes you can have return statement, but it's not the same as other methods in that it doesn't return value, and can be used to exit the method.
6. Returning a value in void method can not be done because it's "void" method.
7. Arguments are passed to a method by its value, so it's a copy of the value rather than the original argument being passed. It's like branching off, modifying it, then returning a different value. It doesn't modify the original value.