Day 6: Terminology Review

Please help NCC conserve resources by not printing this document on campus.

Use the terms in the boxes to fill in the blanks. Terms might be used more than once, or not at all.

~~Accessor Method~~	Actual Parameter	Application Class
Assignment Statement	Class Method	Data Type
Declaration Statement	~~Default Constructor~~	Formal Parameter
Instance Method	~~Instance Variable~~	Local Variable
`main` Method	Method Call	Method Definition
Mutator Method	Object	Parameterized Constructor
Primitive Variable	Programmer-Defined Class	Reference Variable
~~Return Statement~~	~~Return Type~~	`toString` Method

1. A(n) _Access Method_ allows the application class to retrieve the value of an instance variable.

2. A(n) _Default Constructor_ is called when an object is instantiated and there are no arguments to be sent.

3. A(n) _Instance Variable_ stores the data for an object.

4. The _main method_ is where the program begins execution.

5. The _toString method_ always returns a reference to a `String` that contains the state of the object.

6. A(n) _Actual Parameter_ is the value sent to a method.

7. The memory location that stores a value sent to a method is called a(n) _Formal Parameter_

8. A variable that is declared inside a method and can only be used within that method is called a(n) _local variable_

9. A(n) _Method Call_ tells the computer to execute a method.

10. An instance method must be called on a(n) _Object_ .

11. In a method header, the _return type_ specifies the type of data the method will send back to the caller.

12. A(n) _return statement_ sends a value back to the caller.

13. A(n) **Mutator Method** allows the application class to change the value stored in an instance variable.

14. The **data type** specifies what information can be stored in a variable and what operations can be performed on it.

15. A(n) **reference variable** stores the address of an object.

16. The **method definition** contains the method header and the method body.

17. Instance variables are found in the **programmer.defined class**

18. A call to the `toString` method would appear in the **main method**.

19. A(n) **formal parameter** appears inside the parentheses of a method header, and contains a data type and a name.

20. A(n) **Assignment Statement** stores a value in a variable.

21). Scope portion of code where you are allowed to use code { }

object oriented
• code reuse

If((1.equalsuit) && (C1.equassuit(C3)&&
(C1.equalsuit(C4) && ((1.equalsuit ((5))
flush=True

else
flush=False 2