Assignment-6

1.What is private access specifier?

Ans: Private access specifier is accessed only within the class.
It cannot be accessed from outside the class.

- o The access level of a default modifier is only within the package.
- o It cannot be accessed from outside the package.
- o If you do not specify any access level, it will be the default.
- o Private access specifier allows a class to hide its member variables and member functions from other functions and objects.
- o Only functions of the same class can access its private members.
- o Even an instance of a class cannot access its private members.

2. what are getter and setter methods? why do we need them?

Ans: Getter and setter methods are used to protect the data, particularly when creating the class.

For each instance variable, a getter method returns its value while a setter method sets or updates its value. Given this, getters and setters are also known as accessors and mutators respectively. Getters and setters allow you to control how important variables are accessed and updated in your code. Getter and setter methods are used to secure the data.

Example:

```
      public class Vehicle {
 private String fruit;
   // Getter method
 public String getFruit() {
   return fruit;
 }
 // Setter
 public void setFruit(String f) {
   this.fruit = f;
 }
}
```

3.why this keyword in the setter method?

Ans: The this keyword is used to refer to the current object.

Example:

```
   private int id;

public void setID(int ID) {
   this.id = ID;
}

public void getID() {
   return id;
}
```

4.difference between localvariable and member variable/instance variable.
Ans:
Local Variable :
> They are defined as a type of variable declared within programming blocks or subroutines.
> It is important to initialize local variables before use.
> It does not include any access modifiers such as private, public, protected, etc.
> Its access is limited to the method in which it is declared.
> These variables are destroyed when the constructor or method is exited.

Example:
```
int area()
{
	int length = 10; // Local variable
	int breadth = 5; // Local variable
	int rectarea = length*breadth;
	return rectarea;
}
```

Instance Variable:
> They are defined in class but outside the body of methods
> It is not compulsory to initialize instance variables before use
> It includes access modifiers such as private, public, protected, etc.
> It can be accessed throughout the class.
> These variables are destroyed when the object is destroyed.

Example:
```
class Taxes
{
  int count; // Count is an Instance variable
  /*...*/
}
```

5.what is reference variable?
Ans: Reference variables are the used in creating of an object A variable that holds reference of an object is called a reference variable. Variable is a name that is used to hold a value of any type during program.

6.syntax of creating an object?
Ans: Classname reference variable= new Classname();

7.explain in detail what happens when we create an object?
1. Memory is allocated.
2. Fields are initialized to their default values.

3. The "first line" of the chosen constructor is invoked, unless it's an Object.
4. The instance initializer is executed and the fields are initialized to their requested values (actually field initialization is usually compiled as an inline part of the instance initializer).
5. The rest of the constructor code is executed.

8.what is class?

- Class is nothing but a user-defined datatype.
- Class a blueprint of an object.
- Class is a logical entity to create objects that share common properties and methods.
- Class does not have a memory.
- Syntax of class:

```
  Class class_name
{
Field;
Methods;
}
```

9.what is object?

- An object is a instance of class.
- Object is a physical entity/real time entity.
- Object has memory.
- An object is a real-world entity.
- An object is a runtime entity.
- The object is an entity which has state and behavior.
- Syntax of creating an object:

```
ClassName object = new ClassName();
```

10.what are the default values of all the datatypes?

Ans: Int- 0
Float-0.0f
Boolean- false
Byte-0
Short-0
Long-0L
Double-0.0d
Char-'\u000'
String-null

11.difference between the static methods and instance method?
Ans:
Static method
> Static methods can be called without the object of the class.
> Static methods are associated with the class.
> Static methods can only access static attributes of the class
> A static method is declared with the static keyword.
> The static method will exist as a single copy for a class.
> Static members are independent of the objects of the class.

Instance method

> Instance methods require an object of the class.
> Instance methods are associated with the objects.
> Instance methods can access all the attributes of the class.
> Instance methods do not require any keyword.
> The instance method exists as multiple copies depending on the number of instances of the class.

12.Syntax of accessing the member variable in the main?
Ans: Syntax:
```
      Classname object=new classname();
      object. variablename;
```

13.Syntax of instance method defination?
Ans: Syntax:
```
      Access_modifier return_type method Name()
{
   //method body

}
```
Here,
Access_modifier: It defines the access type of method and it is optional to use.
Return_type: Method may return any value. Ex:int , float, string, void etc
Method_name: we can write anything as like a variable name.

14.syntax of static method defination?
Ans: Syntax:
```
      Access_specifier static return_type method Name()
{
   //method body

}
```

15.difference between actual parameter and formal parameter?
Ans:
Actual parameter
> The Actual parameters are the values that are passed to the function when it is invoked.
> In actual parameters, there is no mentioning of data types. Only the value is mentioned.
> The actual parameters are passed by the calling function.
> Those parameters which are addressed in a function call are called actual parameters.
>These can be variables, constants, and expressions, without data types.
Actual parameters are the values referenced in the parameter index of a subprogram call.
> In actual parameters, there is no requirement to define datatype.

Formal parameter
> formal parameters are in the called function.
> The Formal Parameters are the variables defined by the function that receives values when the function is called.
> In formal parameters, the data types of the receiving values should be included
> Parameters addressed in the function description are called formal parameters.
> Formal parameters are variables with the data type.
> Formal parameters are the values referenced in the parameter index of a subprogram.
> In formal parameters, it is mandatory to define the datatype of the receiving value.

16.why we need the parameter or arguments to the methods?
Ans: when the function is called Some values passed with the function these values are called arguments.
A parameter is variable used to define a particular value during a function definition
The parameters and arguments mostly have the same value but theoretically, are diff from each other

17.why we need the return statement and return type to the method?
Ans: A return statement causes the program control to transfer back to the caller of a method.
Every method in java is declared with a return type and it is mandatory for all the java methods. Return statement used to return the value from a method and the flow of program execution comes out of it goes back to the caller method.
• Return type returns a value of expected data type from the method
A return type may be a primitive type like int, float ,double, a reference type or void type.

18.Method can be private.(true or false)
Ans: True, Methods can be private.

19.what is the error message if we access private variable or method out side the class?
Ans: Error message “has a private access in classname ” is a java bug for very very long time.