Assignment No 6

1. What is private access specifier?

- Private access specifier makes its member variables and member functions invisible to other (outer) class, which means the member variables and functions which are declared as private can be accessed only within the class not outside its class.
- Private access specifier allows a class to hide its member variables and member functions from other functions and objects. Only functions of the same class can access its private members. Even an instance of a class cannot access its private members.

2. What are getter and setter methods? Why we need them?

- Getter and Setter are methods used to protect your data and make your code more secure. Getter returns the value (accessors), it returns the value of data type int, String, double, float, etc. For the program's convenience, getter starts with the word "get" followed by the variable name.
- Setter sets or updates the value (mutators). It sets the value for any variable used in a class's programs. and starts with the word "set" followed by the variable name.

3. Why this keyword in setter method?

- The keyword "this" is used in setter method to refer to the current object.

4. Difference between local variable and member/instance variable

- **Local Variables**
  - Local variables are variables which are declared within a method.
  - They won't get any default value and they must be initialized.
- **INSTANCE VARIABLE/MEMBER VARIABLE**
  - Instance variables are variables which are declared inside the class but outside the method
  - Instance variables are part of object
  - Instance variables always get default value.

5. What is reference variable?

- Reference variable is a variable that points to the object created of a given class and allows to access the value of an object.

6. Syntax for creating object

- Using the **new** keyword is the most popular way to create an object or instance of the class. When we create an instance of the class by using the new keyword, it allocates memory (heap) for the newly created **object** and also returns the **reference** of that object to that memory. The syntax for creating an object is:

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

7. Explain in details what happens when we create an object

- When we create an instance of the class by using the new keyword, it allocates memory (heap) for the newly created **object** and also returns the **reference** of that object to that memory.
- When we create an object , memory is allocated to object to hold the properties of the object and also the reference is created which points to the that memory location.

8. What is class?

- A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

9. What is object?

- Object is an instance of a class and it is real time entity.

10. What are the default values of all data types?

- Data type Default value

  - boolean false
  - char '\u0000'
  - byte 0
  - int 0
  - short 0
  - float 0.0f
  - long 0L
  - double 0.0d
  - String null

11. Difference between static methods and instance methods

- Static methods: static methods do not need object creation to be called. they can be called with class name or by the method name itself.

  Instance methods: Instance methods are the methods which are not declared as static. they can be called only with the help of object.

12. Syntax of accessing member variable in main method.

- object.variablename;
  Example: student.studentName;

13. Syntax of instance method definition

- Access _specifier return_ type methodName() { method body; }

• Access specifier - specifies the scope of the method that is who can access this method.

• return_type - returns a value of specified data type from the method.

• methodName - should be given as per the purpose and should following method naming conventions.

method body - set of statements performing particular task can be given in the

method body.

14. Syntax of static method definition

➢ Access _modifier static return_type methodName() { method body; }

• Access specifier -- specifies the scope of the method that is who can access this method.

• static - it is non-access modifier and makes the method to be accessed without creating object.

• return_type - returns a value of specified data type from the method.

• methodName--should be given as per the purpose and should following method naming

conventions.

• method body-- set of statements performing particular task can be given in the method body.

15. Difference between actual parameter and formal parameter

• Actual parameters are those parameters that are specified in the calling function.

• Formal parameters are those parameters that are declared in the called function.

16. Why we need the parameter or arguments to the methods?

• We need parameters to the methods to an input to the method.

• It is necessary to pass the data to methods that are working with data.

17. Why we need the return statement and return type to the method?

➢ 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

18. Method can be private: True or false

➢ True.

19. What is the error message we get if we access private variable or method outside the class?

➢ The filed Class.variable is not visible; Example:The field Employee empId is not visible;