Introduction

- The word “Variable” is a combination of "vary + able" that means its value can be changed.
- Variable is an identifier that denotes a storage location used to store a data value.
- In other words, it is a name of memory location.
- We must declare all variables before using them in program.

Syntax
datatype variable_name = value;

Example

- int a, b, c;   // Declares three int variables, a, b, and c.
- int a = 10, b = 10; // Declares and initialize variables a and b.

Rules for defining variables

- It must start with either alphabets or underscore (_) or dollar ($) symbol.
- It can be combination of alphabets (A-Z, a-z), digits (0-9), underscore and dollar.
- White space is not allowed.
- It should not be a keyword.
- It can be of any length.
- Uppercase and lowercase characters are distinct.
- It can be the name of class or interface. For example
    - int Runnable=10;   //name of interface
    - int String=20;   //name of class

Types of variables

1. Local variables
2. Instance variables
3. Class/Static variables

Local Variables

- They are declared inside the body of methods, constructors, or blocks.
- The scope of local variables is within the body of methods, constructors, or blocks.
- Access modifiers are not used for local variables.
- It is only visible within the declared method, constructor, or block.
- There is no default value for local variables so it must be declared and initialise before the first use.

Instance Variables

- They are declared inside the class but outside the body of the method.
- It is called instance variable because its value is instance specific and is not shared among instances.
- They are created when an object is created and destroyed when the object is destroyed.
- Access modifiers can be used for instance variables.
- They have default values.

  - Numbers : 0
  - Booleans : false
  - Object references : null
- Its values can be assigned during the declaration or within the constructor.
- It is not declared as static.

Class/Static variables

- It is with the static keyword in a class, but outside a method, constructor or a block.
- Only a single copy of static variable is created and shares among all the instances/objects of the class.
- Memory allocation for static variable happens only once when the class is loaded in the memory.
- It cannot be local.
- It is created when the program starts and destroyed when the program stops.
- They have default values.
  - Numbers : 0
  - Booleans : false
  - Object references : null

Example

```
class A
{
        int a=10;               //instance variable
        static int b=20;        //static/class variable
        void display()
        {
                int c=30;       //local variable
        }
}
```

Questions asked in semester paper

No Questions