Assignment - 1

Ques-1 : Analyze the behaviour in the named function in expression.

Soln :

```
var y = function xyz() {
  console.log("Hello, World!");
}
```

Here,

1- y is the variable holding the function,

2- inside the function expression, xyz is the named function.

3- we can only access xyz inside the function body, and not from the outer or global scope.

```
var y = function xyz() {
  console.log("Hello, World!");
}
y();
```

It is valid.

```
var y = function xyz() {
  console.log("Hello, World!");
}
xyz(); //reference error: xyz is not defined
// This will throw an error because xyz is not defined in the global scope
```

It is invalid.

Uses of named function :-

1- for better debugging (stack traces)

2- better in reusability and hoisting

Ques-2 : What is a garbage collector in js?
Soln : It is basically a built-in JavaScript function which is responsible for automatically releasing the memory by eliminating data that is no longer required or accessible.

Working of garbage collector (GC) :-
1- Tracks references,
   It keeps the tracks of those objects/variables which are reachable.
2- Mark and sweep,
   It marks the reachable data and sweeps the unreachable data,which has no reference.
3- The GC is run by the js engines like V8 in the background regularly.

```
function showName(){
    let name = "Yateesh";
}
showName(); //function runs here.
//After successful execution, 'name' is not reachable
//and will be garbage collected.
```

Ques-3 : Event listeners, why is it slow and its solution?
Soln : Event listener is an inbuilt javascript function that waits for a specific action like click, scroll or keypress on a webpage and runs the code when that event happens.

```
element.addEventListener("event", callback);
// The callback function is executed when the event occurs.
```

Why is it slow?
1- too many listeners
2- due to heavy logic inside a listener
3- high frequency events like click, scroll.

Solution:
We should use ‘closures’ in event listeners to overcome these problems.