

Name :: Awaiz Ghafar
ID :: 15269
Dep. :: Cs.
Paper :: OOP (theory)
Teacher :: M. Ayoub Ichan.

①

(Q: NO: 1)

There are three types of variable:
(1) (Local variable)

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

2 - (Instant variable)

A variable declared inside the class but outside the body of the method is called instance variable. It is not declared as static.

3 - (Static variable)

A variable which is declared as static is called static variable.

It can not be local. You can create a single copy of static and share among all the instances of the class.

(2)

Memory allocation for static variable happens only once when the class is loaded in the memory.

Example of Local variable:

```
Public class Test {  
    public void pupage () {  
        int age = 0;  
        age = age + 7;  
        System.out.println ("pupp age  
is : " + age);  
    }  
    public static void main (string args[]) {  
        Test test = new Test ();  
        test.pupage ();  
    }  
}
```

← ↓ →
out Put
Puppy age is : 7.

(3)

(example of instant variable.)

```
Public class Record {
```

```
    Public String name; // this instance  
    variable is visible for any  
    child class.
```

```
    Private int age; // this instance age  
    variable is visible in Record class  
    only.
```

```
Public Record (String Rec name  
{
```

```
    name = Rec name;
```

```
}
```

```
Public void setAge (int Rec sal)
```

```
{
```

```
    age = Rec sal;
```

```
}
```

```
Public void print Rec()
```

```
{
```

```
    System.out.println ("name: "+ name); //
```

```
    print the value for "name"
```

```
    System.out.println ("age: "+ age); //
```

```
    prints the value for "age".
```

```
}
```

(4)

```
public static void main(String args[])  
{  
    Record r = new Record("Ram");  
    r.set age (23);  
    r.print Rec();  
}
```

← ↓ →
out put.
name : Ram
age : 23
← →

(Example of static variable)

```
class variable Demo  
{
```

```
    static int count = 0;
```

```
    public void increment()
```

```
{
```

```
    count ++;
```

```
}
```

```
public static void main(String args[])
```

(5)

```
{
```

```
variable Demo obj1 = new variable Demo();
```

```
variable Demo obj2 = new variable Demo();
```

```
obj1.increament();
```

```
obj2.increament();
```

```
system.out.println("obj1: count is =
```

```
" + obj1.count);
```

```
system.out.println("obj2: count is = " + obj2.  
count);
```

```
}
```

```
}
```

(Out Put)

obj1: count is = 2

obj2: count is = 2

6

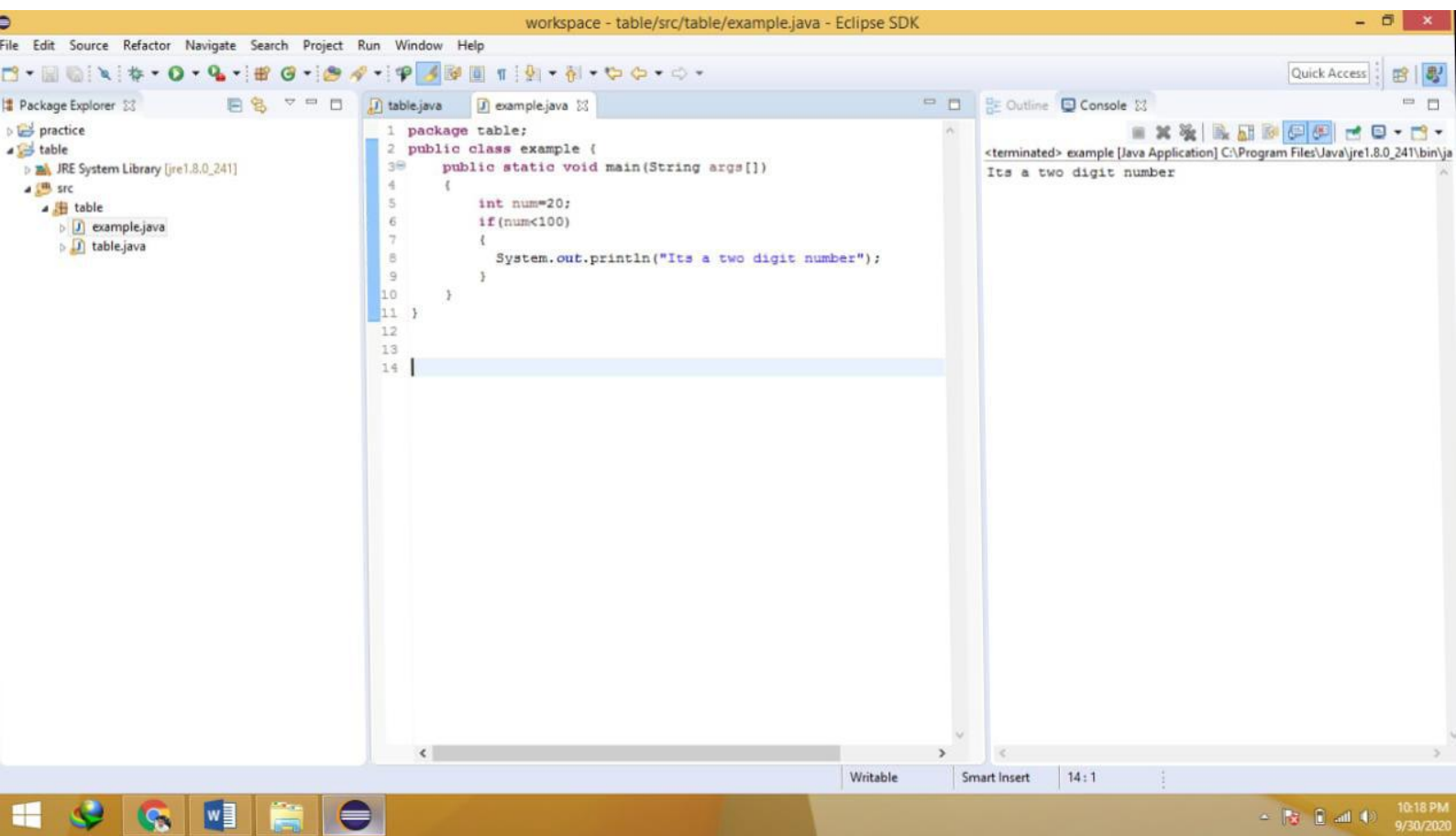
Q: NO: 2.

(If - statement)

If statement consists a condition followed by statement or set of statements as shows below

```
if (condition) {  
    statement (s);  
}
```

The statements gets executed only when the given condition is true. If the condition is false then the statements inside if statement body are completely ignored.



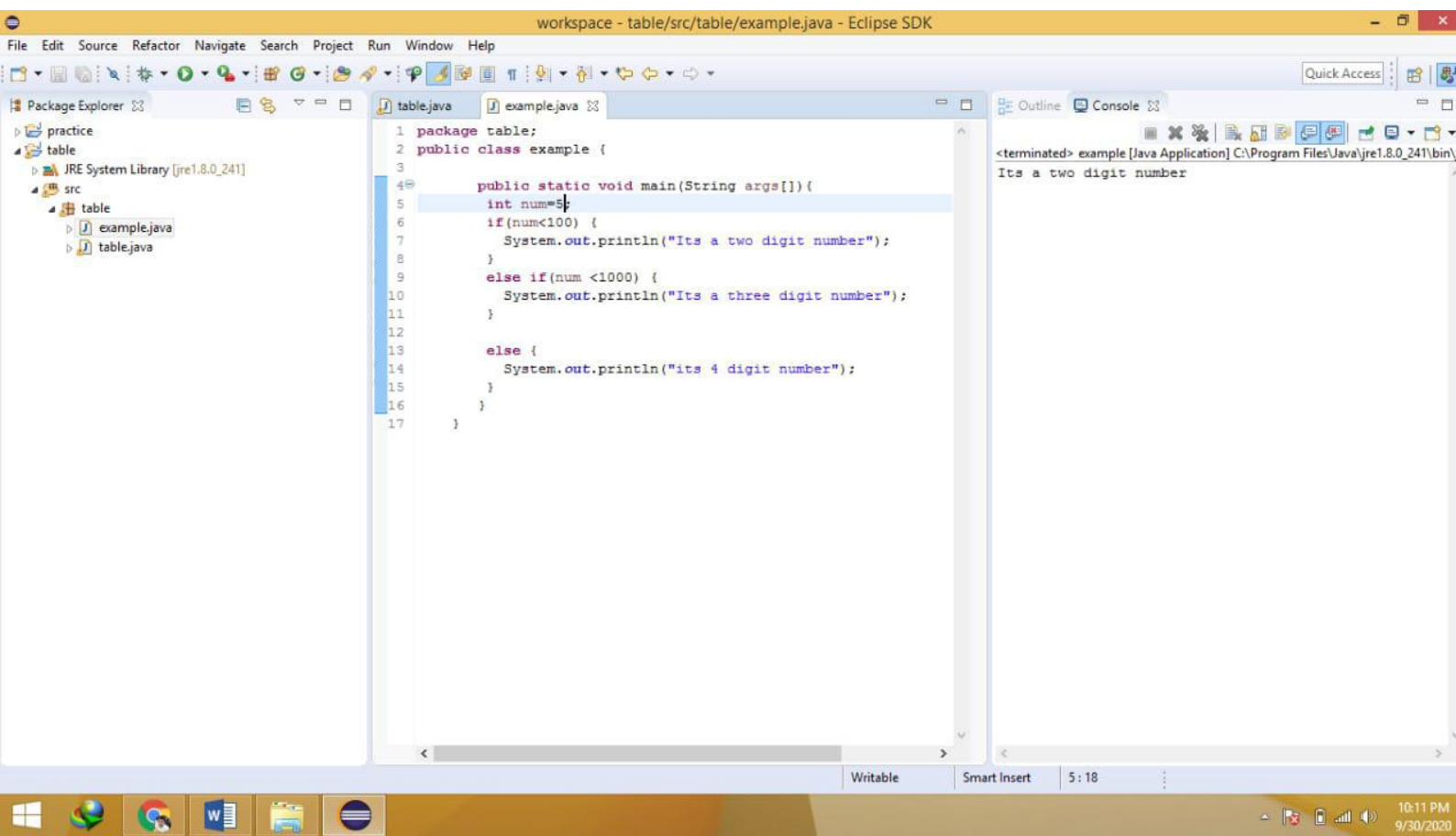
8

(Q: NO: 3)

(IF - else - if statement)

If - else - if statement is used when we need to check multiple conditions. In this statement we have only one "if" and one "else", however we can have multiple "else if". It is also known as if - else - if.

The most important point to note here is that in if - else - if statement, as soon as the condition is met, the corresponding set of statement get executed, rest gets ignored. If none of the condition is met then the statements inside "else" gets executed.



(10)

Q: No: 4.

Loops in Java.

In Programming languages loops are used to execute a set of instruction function repeatedly when some condition become true

Three types of loops in Java

1- (For Loop)

Intro

The Java for loop is a control flow statement that iterates a part of the programs multiple times.

Use

If the number of iteration is fixed it is recommended to use for loop.

Syntax

```
for (init; condition; incr/decr) {  
  // code to be executed  
}
```

(11)

example

```
// for loop
for (int i = 1; i <= 10; i++) {
    System.out.println(i);
}
```

synatan for infinite loop

```
for (; ; ) {
    // (code to be executed)
}
```

←—————→
2- while loop)

intro

The java while loop is a control flow statement that executes a part of the program given boolean condition

use

If the number of iteration is not fixed it is recommended to use while loop.

(12)

Syntax

```
while (condition) {  
    // code to be executed  
}
```

Example

```
// while loop  
int i = 1;  
while (i <= 10) {  
    System.out.println(i);  
    i++;  
}
```

Syntax for infinite loop

```
while (true) {  
    // code to be executed  
}
```

3- do-while loop.

intro.

The Java do while loop is a control flow statement that executes a part of the program at least once and the further execution depends upon the given boolean condition.

Use.

If the number of iteration is not fixed and you must have to execute the loop at least once. It is recommended to use the do-while loop.

Syntax.

```
do {
```

```
    // code to be executed
```

```
} while (Condition);
```

Syntax for infinite loop.

```
do {
```

```
    // code to be executed
```

```
} while (true);
```

14

Example

```
// do-while loop
```

```
int i = 1;
```

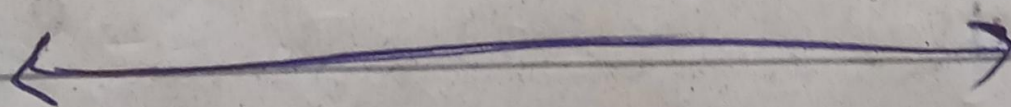
```
do {
```

```
    System.out.println(i);
```

```
    i++;
```

```
}
```

```
while (i <= 10);
```



Q: NO: 5.

