JA111 Day-3

```
class A{
        public static void main(String args[]){
                int a = 10;
                if(true){
                        int a = 5;
                        System.out.println(a);
                }
                System.out.println(a);
        }
}
```

RIGHT ANSWER: Error

in java, you cannot define variable in inner block that has name same as variable of outer block

```
class A{
        public static void main(String args[]){
                int a;  //local variable
                System.out.println(a);
        }
}
```

RIGHT ANSWER: Error

in java, local variable do not have any default value. You have tp provide value before using variable.

```
class A{
        public static void main(String args[]){
                int a = 10;
                if(true){
                        int b = 5;
                        System.out.println(a);
                        System.out.println(b);
                }
                System.out.println(a);
                System.out.println(b);
        }
}
```

RIGHT ANSWER: 10 5 10 Error

Variables of outer block are accessible to inner block butvice versa does not hold.

```
class A{
        public static void main(String args[]){
                System.out.println(a);
                int a = 10;
        }
}
```

RIGHT ANSWER: Error

labelled break

```
class A{
        public static void main(String args[]){
                A:{
                        B:{
                                for(int i = 0; i <= 5; i+=2){
                                        if(i == 4)
                                                break B;
                                        System.out.println(i);
                                }
                                System.out.println(â€œ0â€ );
                        }
                        System.out.println(â€œ1");
                }
                System.out.println(â€œ2â€ );
        }
}
```

RIGHT ANSWER: 0 2 1 2

loop label

```
class A{
        public static void main(String args[]){
```

```
                A: for(int i = 1; i <= 3; i++){
                        B: for(int j = 1; j <= 3; j++){
                                if(i == j)
                                        continue A;
                                System.out.println(i + j);
                        }
                }
        }
}
RIGHT ANSWER: 3 4 5
i = 1
 j = 1
i = 2
 j = 1, 2
i = 3
 j = 1, 2, 3
i = 4 (F)
class A{
        public static void main(String args[]){
                for(int i = 1; i <= 3; i++){
                        for(int j = 1; j <= 3; j++){
                                if(i == j)
                                        continue;
                        System.out.println(i + j);
                        }
                }
        }
}
RIGHT ANSWER:
i = 1
 j = 1, 2
int a = 10;
if(a  == 20)
        System.out.println(â€œAâ€ );
        System.out.println(â€œBâ€ );
is same as
int a = 10;
if(a  == 20){
        System.out.println(â€œAâ€ );
}
System.out.println(â€œBâ€ );
if programmer is not specifying the body of if/else/while/for/do then it is automatically assumed up
to first ;/
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-=-=-=-=-=-=-=-=-=
defining an entity: fields + behaviour
fields are represented using variables
operations are represented using methods
Ceiling Fan
no of blades
color
model
company
size
rpm
turned_on_off
price
operations
turning off
turning on
increasing the speed
decreasing the speed
installation
uninstallation
Havells, Nicola
```

```
package com.masai.class_object_demo;
public class Rectangle {
        double breadth;
        double length;
}
package com.masai.class_object_demo;
public class RectangleDemo {
        public static void main(String[] args) {
                Rectangle rectOne;      //this is reference variable
                rectOne = new Rectangle();      //object is created

                //accessing variables of class
                rectOne.breadth = 10.5;
                rectOne.length = 12.5;

                //creating reference variable and object in one line
                Rectangle rectTwo = new Rectangle();
                rectTwo.breadth = 15.5;
                rectTwo.length = 23.5;

                System.out.println(â€œThe area of rectOne is â€  + (rectOne.breadth *
rectOne.length));
                System.out.println(â€œThe area of rectTwo is â€  + (rectTwo.breadth *
rectTwo.length));
        }
}
output
The area of rectOne is 131.25
The area of rectTwo is 364.25
stack            heap
|768765335|      |___10.5___|__12.5___|
rectOne          breadth    length (768765335)
|768765433|      |___15.5____|__23.5___|
rectTwo          breadth    length (768765433)
as the main is completed
                 |___10.5___|__12.5___|
                 breadth    length (768765335)
                 |___15.5____|__23.5___|
                 breadth    length (768765433)
program ---> execution ---> process
process occupy some memory is RAM this space is called address space.
address space
   data segment
      contains all variables
      data segment is further divided into parts
      1. stack: used to contains local variable
      2. heap: contains the objects and array
      3. metaspace: contains the static variables
   code segment
       contains all methods
static memory allocation
- Managed by compiler i.e. memory allocation and deallocation is manged by compiler itself.
- not flexible i.e. once you have allocated some memory then you can not increase or decrease the
memory size;
- fast
- memory is allocated in the stack area
- in stack memory area, a memory location can have name thatâ€™s why all local variables are created
in stack
dynamic memory allocation
- managed by user (may be by programming language also) i.e. memory allocation is done by user but
user may have to done memory deallocation also.
- flexible i.e. once you have allocated some memory then you can increase or decrease the memory
size;
- slow
- memory is allocated in the heap area
```

- in heap memory area, a memory location cannot have name thatâ€™s why you need to create a reference variable to point to that memory in heap i.e. memory of heap is accessed using some variable of stack.

```
Rectangle rectThree = new Rectangle();
rectThree.length = 20.0;
rectThree.breadth = 10.0;
Rectangle rectFour = rectThree;
stack            heap
|768767878|      |___10.0___|__20.0___| (ref count: 2)
rectThree          breadth    length (768767878)
|768767878|
rectFour
rectFour.length = 15.0;
rectFour.breadth = 13.0;
stack            heap
|768767878|      |___15.0___|__13.0___| (ref count: 2)
rectThree          breadth    length (768767878)
|768767878|
rectFour
rectThree = null;
stack            heap
|null|           |___15.0___|__13.0___| (ref count: 1)
rectThree          breadth    length (768767878)
|768767878|
rectFour
as main method completes execution
stack            heap
                 |___15.0___|__13.0___| (ref count: 0)
                 breadth    length (768767878)
```

A reference variable can point to one object only yet one object can be pointed by multiple reference variables.

parameter passing in the method

1. call by values

```
class A{
        void show(int b){
                b = b + 10;
        }
        public static void main(String args[]){
                A a = new A();
                int i = 20;
                System.out.print(i + â€  â€œ);
                a.show(i);
                System.out.print(i);
        }
}
stack    heap
|8787|   |_____|
  a        (1)  8787
| 20 |
  i: main
20 20
class P{
        int i, j;
        void display(){
                System.out.println(i + â€  â€  + j);
        }
}
class Q{
        void change(P temp){
                temp.i = temp.i + 10;
                temp.j = temp.j + 20;
        }
        public static void main(String args[]){
                P p = new P();
                p.i = 10;
```

```
                p.j = 20;
                p.display();
                Q q = new Q();
                q.change(p);
                p.display();
        }
}
stack      heap
|65765|    |__20__|_40___| (ref count: 2)
  p          i     j 65765
|6567|    |____|
  q          1  6567
10 20
20 40
```