

## Bài tập tại lớp 3

-----  -----  
Tên Sinh Viên \_\_\_\_\_ Lớp \_\_\_\_\_ Mã SV \_\_\_\_\_  
Ngày \_\_\_\_\_ Điểm \_\_\_\_\_

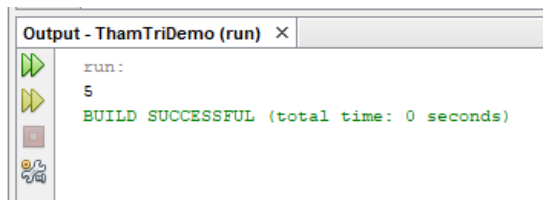
### BÀI 1: Viết chương trình cho phép xuất ra màn hình với định dạng như sau

Ví dụ về khai báo lớp OtoDemo dùng để tạo đối tượng Ô tô

**Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp**

```
12 public class OtoDemo {  
13     _____;  
14     _____;  
15     _____;  
16     String mau;  
17  
18     OtoDemo(_____  
19     {  
20         this.nam = nam;  
21         this. _____;  
22         this. _____;  
23         this.mau = mau;  
24     }  
25  
26     void khoiDong()  
27     {System.out.println("O to " + model + " khoi dong");}  
28  
29     void dung()  
30     {_____ "O to dung");}  
31  
32     void phanh()  
33     {System.out.println("O to phanh");}  
34  
35     void printAll()  
36     {System.out.println("O to " + model + " mau " + mau);}  
37 }
```

### BÀI 2: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



**Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp**

```
12 public class ThamTriDemo {  
13  
14     static void addOneTo _____ {  
15         num = _____
```

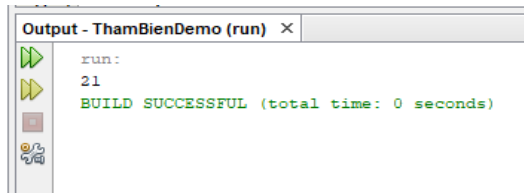
```

16 }
17
18 public static void main(String[] args) {
19     // TODO code application logic here
20     int x = 5;
21     _____ (x);
22     System.out.println(x);
23 }
24 }

```

---

### BÀI 3: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



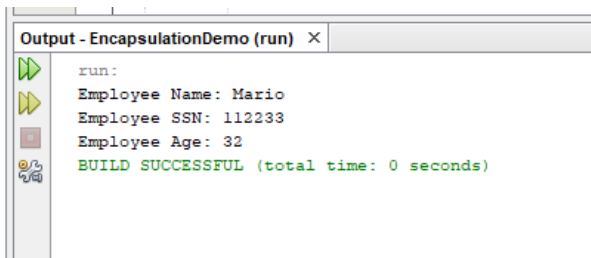
### Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```

12 class Person {
13     private String Name;
14     private int Age;
15
16     public Person(String name) {
17         _____
18     }
19
20     public void setAge(int age) {
21         this.Age = age;
22     }
23
24     public int getAge() {
25         return this.Age;
26     }
27
28 }
29 public class ThamBienDemo {
30
31     /**
32      * @param args the command line arguments
33      */
34     static void celebrateBirthday(Person p) {
35         _____ + 1);
36     }
37     public static void main(String[] args) {
38         // TODO code application logic here
39         Person j;
40         j = new Person("John");
41         j. _____ (20);
42         _____ (j);
43         System.out.println(j. _____ ());
44     }
45 }

```

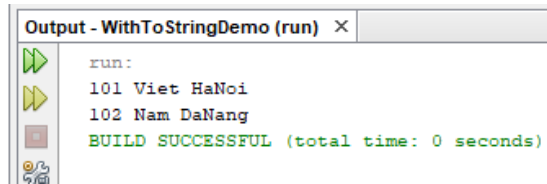
### BÀI 4: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



**Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp**

```
6 package encapsulationdemo;
7
8 class Encapsulation{
9     _____;
10    _____;
11    private int empAge;
12
13    //Getter and Setter methods
14    public int getEmpSSN(){
15        _____
16    }
17
18    public String getEmpName(){
19        _____
20    }
21
22    public int getEmpAge(){
23        return empAge;
24    }
25
26    public void setEmpAge(int newValue){
27        _____;
28    }
29
30    public void setEmpName(String newValue){
31        empName = newValue;
32    }
33
34    public void setEmpSSN(int newValue){
35        _____;
36    }
37 }
38 public class EncapsulationDemo {
39
40     /**
41      * @param args the command line arguments
42      */
43     public static void main(String[] args) {
44         // TODO code application logic here
45         Encapsulation obj = new Encapsulation();
46         obj.setEmpName("Mario");
47         obj.setEmpAge(32);
48         obj.setEmpSSN(112233);
49         System.out.println("Employee Name: " + _____)
50         System.out.println("Employee SSN: " + _____)
51         System.out.println("Employee Age: " + _____)
52     }
53
54 }
```

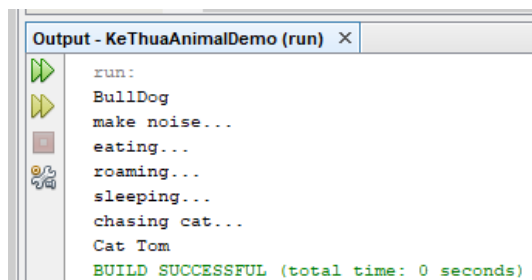
## BÀI 5: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



### Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```
8 class Student {
9     int _____
10    String _____
11    String city;
12    Student(int rollno, String name, String city) {
13        _____
14        _____
15        this.city = city;
16    }
17    public String toString() { //Ghi đè phương thức toString()
18        return rollno + " " + name + " " + city;
19    }
20 }
21 public class WithToStringDemo {
22
23     public static void main(String[] args) {
24         // TODO code application logic here
25         Student s1 = new _____
26         Student s2 = new _____
27
28         _____compiler writes here s1.toString()
29         System.out.println(s2); // compiler writes here s2.toString()
30     }
31 }
```

## BÀI 6: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



### Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```
6 package kethuaanimaldemo;
7
8 class Animal {
9     public String picture;
10    public String food;
11    public boolean hunger;
12    public String boundary;
13    public String location;
```

```

14
15 public Animal(String picture, String food, int hunger, String boundary, String location)
16 {
17     _____;
18     _____;
19     _____;
20     _____;
21     _____;
22 }
23
24 void eat() {
25     System.out.println("eating...");
26 }
27 void makeNoise() {
28     System.out.println("make noise...");
29 }
30 void sleep() {
31     System.out.println("sleeping...");
32 }
33 void roam() {
34     System.out.println("roaming...");
35 }
36 }
37
38 class Dog _____{
39     String chaseCatName;
40     public Dog(String picture, String food, int hunger, String boundary, String location, String chaseCatName)
41     {
42         _____
43         this.chaseCatName=chaseCatName;
44     }
45     public void chaseCat() {
46         System.out.println("chasing cat...");
47     }
48 }
49
50 public class KeThuaAnimalDemo {
51
52     public static void main(String[] args) {
53         // TODO code application logic here
54         Dog d = new Dog_____
55         System.out.println(d.picture);
56         d.makeNoise();
57         d.eat();
58         d.roam();
59         d.sleep();
60         d.chaseCat();
61         System.out.println(d.chaseCatName);
62     }
63

```

**BÀI 7: Viết chương trình cho phép xuất ra màn hình với định dạng như sau**

```
Output - PolymorphismDemo (run) x
run:
Dog make noise...
Cat make noise...
Cat make noise...
Lion make noise...
Cat make noise...
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp**

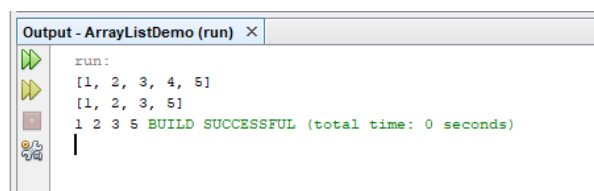
```
12 public abstract class Animal {
13     public _____;
14     public _____
15     public _____
16     public _____;
17     public String location;
18     public Animal()
19     {}
20     public Animal(_____
21         String boundary, String location)
22     {
23         this._____
24         this.food = food;
25         this.hunger = hunger;
26         this. _____;
27         this.location = location;
28     }
29
30     public void eat()
31     {System.out.println("eating...");}
32     public abstract void makeNoise();
33     public void sleep() {
34         System.out.println("sleeping...");
35     }
36     public void roam() {
37         System.out.println("roaming...");
38     }
39 }
40 public class Cat _____{
41     public Cat()
42     {}
43     public Cat(_____,
44         String location)
45     {
46         super(picture,food,hunger,boundary,location);
47     }
48     public void _____{
49         System.out.println("Cat make noise...");
50     }
51     public String toString()
52     { return picture + " " + food + " " + boundary;
53     }
54 }
55
56 public class Dog _____{
57     String chaseCatName;
58     public Dog()
59     {}
60     public Dog(String picture, String food, int hunger, String boundary,
61         _____)
62     {
```

```

63     _____);
64     this.chaseCatName=chaseCatName;
65 }
66 public void chaseCat() {
67     System.out.println("chasing cat...");
68 };
69
70 @Override
71 public void makeNoise() {
72     System.out.println("Dog make noise...");
73 }
74 public String toString()
75 { return picture + " " + food + " " + boundary+" "+chaseCatName;
76 }
77 }
78 public class Lion _____{
79     public Lion()
80     {}
81     public Lion(_____
82         String boundary, String location)
83     {
84         _____);
85     }
86     public void makeNoise() {
87         System.out.println("Lion make noise...");
88     }
89     public String toString()
90     { return _____;
91     }
92 }
93 public class PolymorphismDemo {
94
95     /**
96      * @param args the command line arguments
97      */
98     public static void main(String[] args) {
99         // TODO code application logic here
100        // TODO code application logic here
101        Animal[] kingdom= new _____
102        kingdom[0] = _____
103        _____= new Cat();
104        kingdom[2] = new Cat();
105        _____= new Lion();
106        kingdom[4] = new Cat();
107        for(int i=0;i<kingdom.length;i++)
108            _____makeNoise();
109    }
110 }
111
112 }

```

## BÀI 8: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



```

Output - ArrayListDemo (run) X
run:
[1, 2, 3, 4, 5]
[1, 2, 3, 5]
1 2 3 5 BUILD SUCCESSFUL (total time: 0 seconds)

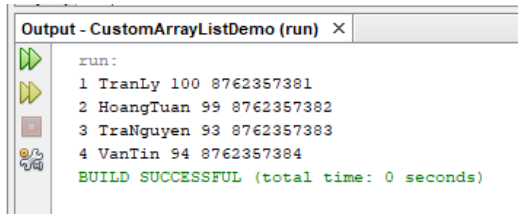
```

### Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```
8 import java.util.ArrayList;
9
10 public class ArrayListDemo {
11
12     public static void main(String[] args) {
13         // TODO code application logic here
14         // size of ArrayList
15         int n = 5;
16
17         //declaring ArrayList with initial size n
18         _____
19
20         // Appending the new element at the end of the list
21         for (int i=1; i<=n; i++)
22             _____
23
24         // Printing elements
25         System.out.println(_____
26
27         // Remove element at index 3
28         arrli. _____
29
30         // Displaying ArrayList after deletion
31         System.out.println(arrli);
32
33         // Printing elements one by one
34         for (int i=0; i<arrli.size(); i++)
35             System.out.print(_____
36     }
37 }
```

---

### **BÀI 8: Viết chương trình cho phép xuất ra màn hình với định dạng như sau**



### Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```
8 import java.util.ArrayList;
9
10 class Customer
11 {
12     // global variables of Customer
13     int roll;
14     String name;
15     int marks;
16     long phone;
17
18     // constructor has type of data that is required
19     Customer(int roll, String name, int marks, long phone)
20     {
21         // initialize the input variable from main
22         // function to the global variable of the class
23         this.roll = roll;
```

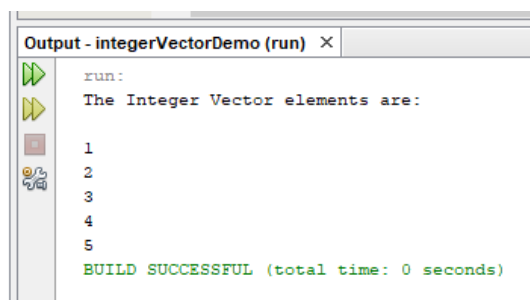


```

24     this.name = name;
25     this.marks = marks;
26     this.phone = phone;
27 }
28 }
29
30 public class CustomArrayListDemo {
31
32     /**
33      * @param args the command line arguments
34      */
35     int n=4;
36     public void addValues(int roll[], String name[], int marks[],
37                           long phone[])
38     {
39         _____
40         for (int i = 0; i < n; i++)
41         {
42             list. _____
43         }
44         printValues(list);
45     }
46
47     public void printValues(ArrayList<Customer> list)
48     {
49         for (int i = 0; i < n; i++)
50         { Customer data = _____
51           System.out.println(data.roll+" "+data.name+" "
52                               +data.marks+" "+data.phone);
53         }
54     }
55     public static void main(String[] args) {
56         // TODO code application logic here
57         int roll[] = {1, 2, 3, 4};
58         String name[] = {"TranLy", "HoangTuan", "TraNguyen", "VanTin"};
59         int marks[] = {100, 99, 93, 94};
60         long phone[] = {8762357381L, 8762357382L, 8762357383L,
61                        8762357384L
62                        };
63
64         // Create an object of the class
65         CustomArrayListDemo custom = new CustomArrayListDemo();
66         custom. _____
67     }
68
69 }

```

### **BÀI 9: Viết chương trình cho phép xuất ra màn hình với định dạng như sau**



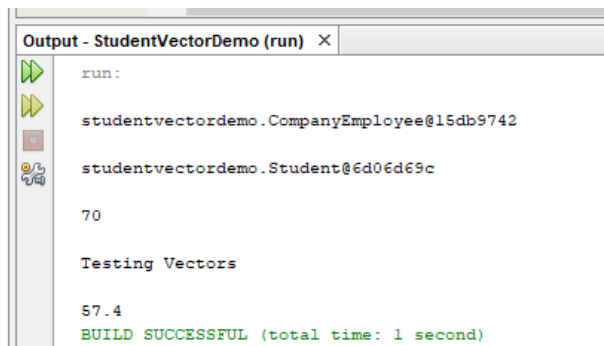
**Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp**

```

8 import java.util.Vector;
9 public class IntegerVectorDemo {
10
11     public static void main(String []args) {
12
13         Vector<Integer> intVector = new Vector<Integer>(); //Declaring a Vector in Java
14         intVector._____
15         intVector._____
16         intVector._____);
17         intVector.add(4);
18         intVector_____
19         //Displaying vector elements
20         System.out.println("The Integer Vector elements are:");
21         System.out.println("");
22         for (int currIntVector : _____) {
23             System.out.println(c_____);
24         }
25 }
26 }

```

### BÀI 10: Viết chương trình cho phép xuất ra màn hình với định dạng như sau



### Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```

9 import java.util.Vector;
10
11 class Student
12 {
13     //data members of class
14     private int rollno;
15     private String name;
16     private String schoolCode;
17
18     //Constructor
19     public Student(int rollno,String name,String schoolcode)
20     {
21         this.rollno=rollno;
22         this.name=name;
23         this.schoolCode=schoolcode;
24     }
25     //putStudent() to print the values of the student Object
26     public void putStudent()
27     {
28         System.out.println("RollNo :"+this.rollno+"\nName :"+this.rollno+"\nSchoolCode :"+this.schoolCode);
29     }
30 }
31
32 //class representing the employee

```

```

33 class CompanyEmployee
34 {
35     //data members of the class
36     public int id;
37     public String name;
38     public long salary;
39     //constructor
40     public CompanyEmployee(int id,String name,long salary)
41     {
42         this.id=id;
43         this.name=name;
44         this.salary=salary;
45     }
46     //putEmployee() to print the values of the employee object
47     public void putEmployee()
48     {
49         System.out.println("Id :"+this.id+"\nName :"+this.name+"\nSalary :"+this.salary);
50     }
51 }
52
53 public class StudentVectorDemo {
54
55     public static void main(String[] args) {
56         // TODO code application logic here
57         Vector vecStuEmp=new Vector(); //Vector class to hold the objects
58         vecStuEmp.add(_____); // adding CompanyEmployee Object to Vector
59         vecStuEmp.add(_____); // adding Student Object to Vector
60         vecStuEmp.add(_____); // adding Integer as Object to Vector
61         vecStuEmp.add(new String("Testing Vectors")); /// adding String as an Object to Vector
62         vecStuEmp.add(_____); /// adding employee as Object to Vector
63
64         //iterating the vector to print the Objects
65         for(Object objStuEmp:vecStuEmp)
66         {
67             /* as Vector holds heterogeneous data objects thus we have to cast the object to its type
68             in order to do this we are using getName() function which gets the name of the class of the given
69             object
70             and compares it with the given class , if it matches then typecast the object to that class */
71             if(objStuEmp.getClass()._____
72             {
73                 System.out.println();
74                 ((CompanyEmployee) _____
75             }
76             else if(objStuEmp.getClass().getName().equals("logicProgramming.Student"))
77             {
78                 System.out.println();
79                 ((Student) _____
80             }
81             // if no match is found that is we will simply print the Object
82             else
83             {
84                 System.out.println();
85                 System.out. _____
86             }
87         }
88     }
89 }
90 }

```