Name : Swapnil Ahire

ID : T090800256

Java Practice Codes

1. Area of Circle

```
package CODES;


public class AREAofCIRCLE {


        public static void main(String[] args) {
          double radius = 5.0;
          double pi = 3.142;


          double area = pi * radius * radius;



          System.out.println("Area of the circle is: " + area);
        }
      }
```

2. Data Type Demo

```
package CODES;


public class DataTypesDemo {
      public static void main (String[]args) {
            int value1 =9/2;
            float value2 =101f/6f;
            double value3 =100d/3d;
```

```
        float value4=100/3;
        System.out.println("value1="+value1);
        System.out.println("value2="+value2);
        System.out.println("value3="+value3);
        System.out.println("value4="+value4);

    }

}
```

3. Char demo

```
package CODES;

public class CHARDEMO {
    public static void main (String[]args) {
        char ch='a';
        System.out.println(ch);

        //assigning number to char
        char ch1=65;
        System.out.println(ch1);

    }

}
```

4. Operator Demo

```
package CODES;


public class OperatorDemo {
	public static void main (String []args) {
		int a=10;
		int b=20;
		int x=10;
		System.out.println("a and b value"+"before the operation :"+a+""+b);
		++a;
		int c=++a + b +a--;
		System.out.println("c value after the operations :"+c);


		int d=c++ + a + b--;
		System.out.println("c value after the operations :"+d);




	}

}
```

5. For Loop

```
package CODES;


public class ForLOOP {
	public static void main (String[]args) {
		for (int i=1;i<=1000;i++)
		{
```

```
                System.out.println("Value of i :");
                System.out.println(i);
            }
        }

}
```

6. Break

```
package CODES;

public class BREAK {
    public static void main(String[] args) {
        for (int i=5;i<10;i++)
        {
            if (i==5)
                break;
            System.out.println(i);
        }
    }

}
```

7. Constructor Java

```
package CODES;

public class Constructor {
```

```
private String customerName;
private int customerId;
private String customerCity;



public Constructor() {
	super();
	// TODO Auto-generated constructor stub
}



public Constructor(String customerName, int customerId, String customerCity) {
	super();
	this.customerName = customerName;
	this.customerId = customerId;
	this.customerCity = customerCity;
}



public String getCustomerName() {
	return customerName;
}



public void setCustomerName(String customerName) {
	this.customerName = customerName;
```

```
}

public int getCustomerId() {
	return customerId;
}

public void setCustomerId(int customerId) {
	this.customerId = customerId;
}

public String getCustomerCity() {
	return customerCity;
}

public void setCustomerCity(String customerCity) {
	this.customerCity = customerCity;
}

@Override
public String toString() {
```

```
		return "Constructor [customerName=" + customerName + ", customerId=" +
customerId + ", customerCity="
				+ customerCity + "]";
	}




		// Constructor are used to assigning values to class variables at the time of object
creation
		// Types : Default and parameterized constructor


	}
```

8. Continue

```
package CODES;


public class Continue {
	public static void main(String[] args) {
		for (int k=5; k<15;k++)
		{
			if (k%2 !=0)
				continue;
			System.out.println(k+" ");
		}
	}

}
```

9. Do While

```
package CODES;


public class DOWHILE {
	public static void main(String[] args) {
		int i =11;
		do {
			System.out.println(i);
			i++;
	}
	  while(i<=100);


}
}
```

10. Encapsulation

```
package CODES;


public class ENCAPSULATION1 {


	private int serialNum;
	private String name;
	private int age;
	public int getSerialNum() {
		return serialNum;
	}
	public void setSerialNum(int serialNum) {
		this.serialNum = serialNum;
```

```
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public int getAge() {
		return age;
	}
	public void setAge(int age) {
		this.age = age;
	}
	@Override
	public String toString() {
		return "ENCAPSULATION1 [serialNum=" + serialNum + ", name=" + name + ", age=" +
age + "]";
	}



}
```

11.Executer :

```
package CODES;


public class Executer {
	public static void main (String []args) {
		//accessing same package class
```

```
		Modifiers b1 = new Modifiers();
		b1.methodDefault();
		b1.methodProtected();
		b1.methodPublic();
		//b1.methodPrivate();
	}

}
```

12.FOR EATCH LOOP

```
package CODES;

public class FOREACHLOOP {
	public static void main(String[] args) {
		int a[]= {10,20,30,40,50};
		for (int i:a) {
			System.out.println(i);
		}
	}

}
```

13.FOR LOOP

```
package CODES;

public class ForLOOP {
```

```
	public static void main (String[]args) {
		for (int i=1;i<=1000;i++)
		{
			System.out.println("Value of i :");
			System.out.println(i);
		}
	}

}
```

14.Modifiers

```
package CODES;

public class Modifiers {
	//public private protected default >>>modifiers

	int varDefault =10;
	public int varPublic=20;
	protected int varProtected=30;
	private int varPrivate=40;

	//declare all access methods
	void  methodDefault()
	{
		System.out.println("Default Access base class");
		System.out.println("Default variable : +varDefault");

	}
	void  methodPublic()
```

```
{
        System.out.println("Public Access base class");
        System.out.println("Public variable : +varPublic");

}
void  methodProtected()
{
        System.out.println("Protected Access base class");
        System.out.println("Protected variable : +varProtected");

}
void  methodPrivate()
{
        System.out.println("private Access base class");
        System.out.println("Private variable : +varPrivate");

}

        }
```

15.Nested For loop

```
package CODES;

public class NESTEDFORLOOP {
        public static void main(String[] args) {
                //program to print multiplication of tables in a given range
                int beg=10;
                int end=20;

                for(int i=beg; i<=end;i++)
```

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

                }


        }

}
```

//for each loop used for it converts array format to normal format

16.PERSON Class

```
package NEWBIGPROGRAMS;

public class PERSON {

        // data members                 // object to string is super parent
        private String name;
        private int income;
        private String gender;
        private int age;
        private int tax;

        // getter and setter //  right click > source > generate select getter and setter
        public String getName() {
```

```
        return name;
}
public void setName(String name) {
        this.name = name;
}
public int getIncome() {
        return income;
}
public void setIncome(int income) {
        this.income = income;
}
public String getGender() {
        return gender;
}
public void setGender(String gender) {
        this.gender = gender;
}
public int getAge() {
        return age;
}
public void setAge(int age) {
        this.age = age;
}
public int getTax() {
        return tax;
}
public void setTax(int tax) {
        this.tax = tax;
}
// object method return string representation //
```

```
	@Override
	public String toString() {
		return "PERSON [name=" + name + ", income=" + income + ", gender=" + gender + ", age=" + age + ", tax=" + tax
				+ ", getName()=" + getName() + ", getIncome()=" + getIncome() + ", getGender()=" + getGender()
				+ ", getAge()=" + getAge() + ", getTax()=" + getTax() + ", getClass()=" + getClass() + ", hashCode()="
				+ hashCode() + ", toString()=" + super.toString() + "]";
	}




}
```

17.Switch

```
package CODES;


public class Swich {
	public static void main(String[] args) {
		char x = 'l';
		switch(x)
		{
		case 'l':
		case 'L':
			System.out.println(x+" is a lion");
			break;


		}
```

```
	}


}
```

18.While

```
package CODES;


public class WHILE {
	public static void main(String[] args) {
		int i=99;
		while (i<=1000) {
			System.out.println(i);
			i++;
		}
	}


}
```