ADDON DAY-3

1)Getting character:

```
import java.util.Scanner;

public class Main
{
	public static void main(String[] args) {
	Scanner d=new Scanner(System.in);
	System.out.println("Enter thechar:");
	char a=d.nextLine().charAt(0);
	System.out.println(a);
	}
}
```

Output:

```
Xyz
X
```

2)if we use take 0 in if statement:

```
import java.util.Scanner;
public class Main
{
	public static void main(String[] args) {
		Scanner a=new Scanner(System.in);
		if(0)
		{
		   System.out.println("hello");
		}
		else{
		    System.out.println("hello all");

		}
	}
}
```

Output:

```
Main.java:13: error: incompatible types: int cannot be converted to boolean
                if(0)
                   ^
```

```
1 error
```

3)using static without creating object:

```
class demo{
   static int a = 90;
   static int v = 87;
   public static void display()
   {
      System.out.println("hello");
   }

}
public class Main
{
	public static void main(String[] args) {
		System.out.println(demo.a);
		demo.display();
	}
}
```

Output:

```
90
hello
```

4)using non static with creating object(regular program):

```
class demo{
   String s1="welcome home";
   void display()
   {
      System.out.println("hello");
   }

}
public class Main
{
	public static void main(String[] args) {
	demo g=new demo();
	System.out.println(g.s1);
	g.display();
	}
```

Output:

```
welcome home
hello
```

CONSTRUCTOR:

*special method or member function must have same name as the class name.

*constructor does not allow access modifiers and also written type.

Syntax:

```
Class demo{
    demo(){
        system.out.println(“hello);
}
}
```

5)constructor simple program:

```
public class Main
{
	public static void main(String[] args) {
		demo f=new demo();
	}
}
class demo{
demo(){
   System.out.println("hello");
}
}
```

Output:

```
hello
```

Explanation: in the demo class we have to mention the print statement the plain class will not run .so, we use to main method and we have to create object using demo class.the output will display on the screen.

6)using constructor declare variable to access main method:

```
public class Main
{
      public static void main(String[] args) {
            demo f=new demo("AUDI",4);
            System.out.println(f.name);
            System.out.println(f.car);
      }
}
class demo{
  String name;
  int car;
demo(String name,int car){
  this.name=name;
  this.car=car;

}
}
```

Output:

```
AUDI
4
```

Explanation: we have to declare variable in demo class not a constructor. then, we have to give argument .use in “this” keyword then main class we have to access the variable use in object in demo class within object calling area argument we have to mention variable value.

7)constructor using Encapsulation:

```
public class Main
{
      public static void main(String[] args) {
            company g=new company("Anushika",60000);
            System.out.println(g.getemployee());
            System.out.println(g.getsalary());
      }
}
class company{
  private String employee;
  private int salary;
  company(String employee,int salary)
  {
     this.employee=employee;
     this.salary=salary;
  }
  public String getemployee()
  {
     return employee;
  }
  public int getsalary()
  {
     return salary;
  }
}
```

Output:

```
Anushika
60000
```

Explanation:constructor class using encapsulation we have declare private variable and constructor class we use “this” keyword in variable then main method we have create class name of object with give parameter.we give variable .only we give private the we have give public using “get”keyword in variable of method then give the “return” keyword of variable.in main class we give object and “get”variablename()calling.

8)update name same 7th program:

```
public class Main
{
        public static void main(String[] args) {
                company g=new company("Anushika",60000);
                System.out.println(g.getemployee());
                System.out.println(g.getsalary());
                g.setemployee("Aarthi");
                System.out.println("Enter the update name:"+g.getemployee());

        }
}
class company{
   private String employee;
   private int salary;
   company(String employee,int salary)
   {
       this.employee=employee;
       this.salary=salary;
   }
   public String getemployee()
   {
       return employee;
   }
   public int getsalary()
   {
       return salary;
   }
   public void setemployee(String employee)
   {
       this.employee=employee;
   }
}
```

```
Output:
Anushika
60000
Enter the update name:Aarthi
```

9)same 8th program just adding extra money:

```
public class Main
{
            public static void main(String[] args) {
                        company g=new company("Anushika",60000);
                        System.out.println(g.getemployee());
                        System.out.println(g.getsalary());
                        g.setemployee("Aarthi");
                        System.out.println("Enter the update name:"+g.getemployee());
                        g.setsalary(1000);
                        System.out.println("Enter the update salary:"+g.getsalary());
            }
}
class company{
    private String employee;
    private int salary;
    company(String employee,int salary)
    {
        this.employee=employee;
        this.salary=salary;
    }
    public String getemployee()
    {
        return employee;
    }
    public int getsalary()
    {
        return salary;
    }
    public void setemployee(String employee)
    {
        this.employee=employee;
    }
    public void setsalary(int cash)
    {
       salary=salary+cash;
    }
}
```

Output:

```
Anushika
60000
Enter the update name:Aarthi
Enter the update salary:61000
```

9.1)same 9th program using if condition:

```
public class Main
{
            public static void main(String[] args) {
                        company g=new company("Anushika",60000);
                        System.out.println(g.getemployee());
                        System.out.println(g.getsalary());
                        g.setemployee("Aarthi");
                        System.out.println("Enter the update name:"+g.getemployee());
                        g.setsalary(2000);
                        System.out.println("Enter the update salary:"+g.getsalary());
            }
}
class company{
    private String employee;
    private int salary;
    company(String employee,int salary)
    {
        this.employee=employee;
        this.salary=salary;
    }
    public String getemployee()
    {
        return employee;
    }
    public int getsalary()
    {
        return salary;
    }
    public void setemployee(String employee)
    {
        this.employee=employee;
    }
    public void setsalary(int cash)
    {
       if(60000>0)
       {
           salary+=cash;
           System.out.println("update salary:"+salary);
       }
    }
}
```

Output:

```
Anushika
60000
Enter the update name:Aarthi
update salary:62000
```

9.2)same 9th program the differs is we just subtract the amount:

```
public class Main
{
            public static void main(String[] args) {
                        company g=new company("Anushika",60000);
                        System.out.println(g.getemployee());
                        System.out.println(g.getsalary());
                        g.setemployee("Aarthi");
                        System.out.println("Enter the update name:"+g.getemployee());
                        g.setsalary(2000);
                        g.setwithdraw(5000);

            }
}
class company{
    private String employee;
    private int salary;
    company(String employee,int salary)
    {
        this.employee=employee;
        this.salary=salary;
    }
    public String getemployee()
    {
        return employee;
    }
    public int getsalary()
    {
        return salary;
    }
    public void setemployee(String employee)
    {
        this.employee=employee;
    }
    public void setsalary(int cash)
    {
       if(60000>0)
       {
           salary+=cash;
           System.out.println("update salary:"+salary);
       }
    }
    public void setwithdraw(int withdraw)
    {
        salary-=withdraw;
        System.out.println("withdraw amt:"+salary);
    }
    }
```

Output:

```
Anushika
60000
Enter the update name:Aarthi
update salary:62000
withdraw amt:57000
```