QUESTION BANK PROGRAMS

Definition 1 : Write a program to define abstract class, with two methods addition() and subtraction(). addition() is abstract method. Implement the abstract method and call that method using a program(s). (Abstract Class Concept)

Program:

```
import java.util.Scanner;

 abstract class Calculation

 {

        double num1,num2;

        abstract double addition(double num1,double num2);

        double subtraction(double num1,double num2)

        {

                return num1-num2;

        }

 }

 class Add extends Calculation

 {

        public double addition(double n1,double n2)

        {

                return n1+n2;

        }

 }

 class Abstract_class
```

```
{
        public static void main(String []args)
        {
                Calculation c1;         // reference of abstract class
                c1=new Add();                           // object of an Add class which      //extends
abstract class Calculation
                Scanner s=new Scanner(System.in);
                System.out.println("Enter two numbers:");
                c1.num1=s.nextDouble();
                c1.num2=s.nextDouble();
                double sum=c1.addition(c1.num1,c1.num2);
                System.out.println("Addition is :" + sum);
                double sub=c1.subtraction(c1.num1,c1.num2);
                System.out.println( "Substraction is :" + sub);

        }
}
```

OUTPUT:

```
C:\Windows\system32\cmd.exe

E:\java\question_bank>javac Abstract_class.java

E:\java\question_bank>java Abstract_class
Enter two numbers:
12
34
Addition is :46.0
Substraction is :-22.0
```

Definition 2 : <u>Write a program that divides two numbers. Handle all exceptions that can be generated in this program. (Exception Handling)</u>

<u>Program:</u>

```
import java.util.*;

import java.io.*;

class Exception_Handling

{

	public static void main(String []args)

	{

		float i,j,k=0;

		i=45;

		Scanner s=new Scanner(System.in);

		try

		{

			System.out.println("Enter a number :");

			j=s.nextFloat();

			k=i/j;

			System.out.println("Division is :" +k);

		}

		catch(ArithmeticException ae)

		{

			System.out.println("Arithmatic Exception" +ae);

		}
```

```
catch(NumberFormatException ne)
{
        System.out.println("Number Format Exception" +ne);
}
catch(InputMismatchException ime)
{
        System.out.println("Input Mismatch Exception" +ime);
}
finally                 // this block execute atleast once // optional no need to implement
{
        System.out.println("Bye");
}
    }
}
```

OUTPUT:

```
C:\Windows\system32\cmd.exe

E:\java\question_bank>javac Exception_Handling.java

E:\java\question_bank>java Exception_Handling
Enter a number :
ty
Input Mismatch Exceptionjava.util.InputMismatchException
Bye

E:\java\question_bank>java Exception_Handling
Enter a number :
12
Division is :3.75
Bye
```

Definition 4: Define time class with hour and minute. Also define addition method to add two time objects. (Class and Object Concept)

Program:

```
import java.util.Scanner;

class Time

{

	int hour,minute;

	Time(int hour,int minute)

	{

		this.hour=hour;

		this.minute=minute;

	}

	void addition(Time obj1,Time obj2)

	{

		int min=obj1.minute+obj2.minute;

		int hr=obj1.hour+obj2.hour;

		if(min>=60)

		{

			hr=hr+(min/60);

			min=min%60;

		}

		System.out.println( hr + " Hours " + " : " + min  + " Minutes") ;

	}
```

```
}

class Time_Hour

{

        public static void main(String []args)

        {

                int h,m;

                Scanner s=new Scanner(System.in);

                System.out.println("Enter hour and minute for object 1: ");

                h=s.nextInt();

                m=s.nextInt();

                Time t1=new Time(h,m);

                System.out.println("Enter hour and minute for object 2: ");

                h=s.nextInt();

                m=s.nextInt();

                Time t2=new Time(h,m);

                t1.addition(t1,t2);

        }}
```

OUTPUT:

```
C:\Windows\system32\cmd.exe

E:\java\question_bank>javac Time_Hour.java

E:\java\question_bank>java Time_Hour
Enter hour and minute for object 1:
2
80
Enter hour and minute for object 2:
3
70
7 Hours   : 30 Minutes
```

Definition 5: Write an application that takes input from command-line argument. If an argument is found that does not begin with an upper case letter,display error message and terminate. (Command line argument and String class Concept)

Program:

```
class Command_Line
{
	public static void main(String []args)
	{
		String str;
			if(Character.isUpperCase(args[0].charAt(0)))
		{
			System.out.println("No error: " +args[0]);
		}
			else
		{
			System.out.println("There is a error in a string : " +args[0]);
		}
	}
}
```

OUTPUT:

```
C:\Windows\system32\cmd.exe

E:\java\question_bank>javac Command_Line.java

E:\java\question_bank>java Command_Line Hi
No error: Hi

E:\java\question_bank>java Command_Line my
There is a error in a string : my
```

Definition 6: Write a program to demonstrate the multipath inheritance for the classes having relations as shown in figure. (Interface Inheritance Concept)

Program:

```
interface A
        {
            public void display_A();
        }
        interface B extends A
        {
            public void display_B();
        }
        interface C extends A
        {
            public void display_C();
        }
        class D implements B,C
        {
            public void display_A()
            {
                System.out.println("This is a class A which is a super class");
            }
            public void display_B()
```

```
        {
            System.out.println("This is a class B which implements interface A");
        }
        public void display_C()
        {
            System.out.println("This is a class C which implements interface A");
        }
        public void display_D()
        {
            System.out.println("This is a class D which implements interface B and
C");
        }
    }
    class Multipath_Inheritence
    {
        public static void main(String []args)
        {
            D d1=new D();
            d1.display_A();
            d1.display_B();
            d1.display_C();
            d1.display_D();
        }  }
```

OUTPUT:

```
C:\Windows\system32\cmd.exe

E:\java\question_bank>javac Multipath_Inheritence.java

E:\java\question_bank>java Multipath_Inheritence
This is a class A which is a super class
This is a class B which implements interface A
This is a class C which implements interface A
This is a class D which implements interface B and C
```

Definition 7: Design a class named Fan to represent a fan. The class contains:

- Three constants named SLOW, MEDIUM and FAST with values 1,2 and 3 to denote the fan speed.

- An int data field named speed that specifies the speed of the fan (default SLOW).

- A boolean data field named f_on that specifies whether the fan is on(default false).

- A double data field named radius that specifies the radius of the fan (default 4).

- A data field named color that specifies the color of the fan (default blue).

- A no-arg constructor that creates a default fan.

- A parameterized constructor initializes the fan objects to given values.

- A method named display() will display description for the fan. If the fan is on, the display() method displays speed, color and radius.

If the fan is not on, the method returns fan color and radius along with the message fan is off.

-Write a test program that creates two Fan objects. One with default values and the other with medium speed, radius 6, color brown, and turned on status true.

-Display the descriptions for two created Fan objects. (Class, Object, conctructor and Method concepts)

Program:

```
import java.util.Scanner;

class Fan

{

	int SLOW=1,MEDIUM=2,FAST=3;

	int speed;

	boolean f_on;

	double radious;

	String color;

	Fan()

	{

		this.speed=SLOW;

		this.f_on=false;

		this.radious=4;

		this.color="blue";

	}

	Fan(int speed,boolean f_on,double radious,String color)

	{

		this.speed=speed;

		this.f_on=f_on;

		this.radious=radious;

		this.color=color;

	}

	String display(String str)
```

```
	{
		return "speed : " + speed +"\n"+ "radious :" + radious +"\n"+ "color :"  + color
+"\n"+str;
	}
}
class Fan_Demo
{
	public static void main(String []args)
	{
		Fan f1=new Fan();
		Fan f2;
		Scanner s=new Scanner(System.in);
		System.out.println("Enter weather the fan is on or off: " + "true=on" + "
false=off");
		f1.f_on=s.nextBoolean();
		if(f1.f_on==true)
		{
			f2=new  Fan(2,f1.f_on,6,"brown");
			System.out.println( "For object 1: " + f1.display("Fan is on"));
			System.out.println( "\n\nFor object 2: " + f2.display("Fan is on"));
		}
		else
		{
```

```
                f1.speed=0;

                f2=new Fan(0,f1.f_on,6,"brown");

                System.out.println("For object 1 :" + f1.display("Fan is off"));

                System.out.println("\n\nFor object 2 :" + f2.display("Fan is off"));

            }

        }

}
```

OUTPUT:

C:\Windows\system32\cmd.exe

```
E:\java\question_bank>javac Fan_Demo.java

E:\java\question_bank>java Fan_Demo

Enter weather the fan is on or off: true=on false=off
true
For object 1: speed : 1
radious :4.0
color :blue
Fan is on


For object 2: speed : 2
radious :6.0
color :brown
Fan is on

E:\java\question_bank>java Fan_Demo
Enter weather the fan is on or off: true=on false=off
false
For object 1 :speed : 0
radious :4.0
color :blue
Fan is off


For object 2 :speed : 0
radious :6.0
color :brown
Fan is off
```