Batch: B1 **Roll No.: 1711072**

Experiment / assignment / tutorial No. 09

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE : PACKAGES

AIM: Write a program to create a package 'myPack' which contains a class Trigonometry. The class contains following static methods.

i) sin() –accepts degree (0,30,60,90)

ii) cos() - accepts degree (0,30,60,90)

iii)tan()- accepts degree (0,30,60,90)

iv)cot()-- accepts degree (0,30,60,90)

v)cosec()-- accepts degree (0,30,60,90)

vi)sec()-- accepts degree (0,30,60,90)

(Do not make use of inbuilt functions. Use the functions of user defined class Trigonometry by importing myPack).

Expected OUTCOME of Experiment:

CO4: Demonstrate programs on interface, exceptions, multithreading and applets.

Books/ Journals/ Websites referred:

1. Balagurusamy, E. Programming With Java:A Primer 3E , Tata McGraw-Hill Education, 2006

2. Herbert Schildt, The Complete Reference JAVA2, 2nd ed., TMH, 2002

Pre Lab/ Prior Concepts:

A **package** is a collection of related classes. It helps **to organize** your classes into a folder structure and makes it easy to locate and use them. More importantly, it helps improve **re-usability.**

. Packages are used for:

1. Preventing naming conflicts. For example there can be two classes with name Employee in two packages, college.staff.cse.Employee and college.staff.ee.Employee.

2. Making searching/locating and usage of classes, interfaces, enumerations and annotations easier

3. Providing controlled access: protected and default have package level access control. A protected member is accessible by classes in the same package and its subclasses. A default member (without any access specifier) is accessible by classes in the same package only.

4. Packages can be considered as data encapsulation (or data-hiding).

Adding a class to a Package: We can add more classes to a created package by using package name at the top of the program and saving it in the package directory. We need a new java file to define a public class, otherwise we can add the new class to an existing .java file and recompile it.

Sub packages: Packages that are inside another package are the subpackages. These are not imported by default, they have to imported explicitly. Also, members of a subpackage have no access privileges, i.e., they are considered as different package for protected and default access specifiers.

Example:

```
import java.util.*;
```

util is a subpackage created inside java package.

To create a package (Example)

Step 1) Copy the following code into an editor

```
package p1;
class c1{
public void m1(){
System.out.println("Method m1 of Class c1");
}
public static void main(String args[]){
c1 obj = new c1();
obj.m1();
}}
```

Step 2) Save the file as Demo.java. Compile the file as, javac – d . Demo.java

Step 3) Run the code as java p1.c1

To create a sub-package

Step1) Copy the following code into an editor

```
package p1.p2;
class c2{
public void m2(){
System.out.println("Method m2 of Class c2");
}
public static void main(String args[]){
c2 obj = new c2();
obj.m2();
}}
```

Step 2) Save the file as Demo2.java. Compile the file as javac – d . Demo2.java

Step 3) Run the code as java p1.p2.c2

Implementation Details:
Program 1: Driver class in same package

```
package myPack;
import java.util.*;
import java.lang.*;
public class Trigonometry{
  static double sin(int x){
    if(x==0)
      return 0;
    else if(x==30)
      return 0.5;
    else if(x==60)
      return 0.866;
    else if(x==90)
      return 1;
    else
      return -1;
  }
  public static double cos(int x){
    if(x==0)
      return 1;
    else if(x==30)
      return 0.866;
    else if(x==60)
```

```
    return 0.5;
  else if(x==90)
    return 0;
  else
    return -1;
}
public static double tan(int x){
  if(x==0)
    return 0;
  else if(x==30)
    return 0.5773;
  else if(x==60)
    return 1.732;
  else if(x==90)
    return -2;
  else
    return -1;
}
public static double cot(int x){
  if(x==0)
    return -2;
  else if(x==30)
    return 1.732;
  else if(x==60)
    return 0.5773;
  else if(x==90)
    return 0;
  else
    return -1;
}
public static double cosec(int x){
  if(x==0)
    return -2;
  else if(x==30)
    return 2;
  else if(x==60)
    return 1.1547;
  else if(x==90)
    return 1;
  else
    return -1;
}
public static double sec(int x){
  if(x==0)
```

```
      return 1;
    else if(x==30)
      return 1/0.866;
    else if(x==60)
      return 2;
    else if(x==90)
      return -2;
    else
      return -1;
  }
}
class Main{
  public static void main(String[] args){
    Scanner sc=new Scanner(System.in);
    int choice, x;
    double result;
    do{
      System.out.print("1. Sine\n2. Cosine\n3. Tangent\n4. Cotangent\n5.
Cosecant\n6. Secant\n7. Exit\nEnter a choice: ");
      choice=sc.nextInt();
      switch(choice){
        case 1:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.sin(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 2:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.cos(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 3:
```

```
      System.out.print("\nEnter value of x(0,30,60,90): ");
      x=sc.nextInt();
      result=Trigonometry.tan(x);
      if(result==-1)
        System.out.println("Enter from (0,30,60,90).");
      if(result==-2)
        System.out.println("Value undefined.");
      else
        System.out.println("Value is: "+result);
      break;
    case 4:
      System.out.print("\nEnter value of x(0,30,60,90): ");
      x=sc.nextInt();
      result=Trigonometry.cot(x);
      if(result==-1)
        System.out.println("Enter from (0,30,60,90).");
      if(result==-2)
        System.out.println("Value undefined.");
      else
        System.out.println("Value is: "+result);
      break;
    case 5:
      System.out.print("\nEnter value of x(0,30,60,90): ");
      x=sc.nextInt();
      result=Trigonometry.cosec(x);
      if(result==-1)
        System.out.println("Enter from (0,30,60,90).");
      if(result==-2)
        System.out.println("Value undefined.");
      else
        System.out.println("Value is: "+result);
      break;
    case 6:
      System.out.print("\nEnter value of x(0,30,60,90): ");
      x=sc.nextInt();
      result=Trigonometry.sec(x);
      if(result==-1)
        System.out.println("Enter from (0,30,60,90).");
      if(result==-2)
        System.out.println("Value undefined.");
      else
        System.out.println("Value is: "+result);
      break;
    case 7:
```

```
          System.exit(1);
        default:
          System.out.println("Enter a valid choice. ");
      }
    }while(choice!=7);
  }
}
```

Program 2: Driver class outside myPack package

myPack Trigonometry class

```
package myPack;
import java.util.*;
import java.lang.*;
public class Trigonometry{
  static double sin(int x){
    if(x==0)
      return 0;
    else if(x==30)
      return 0.5;
    else if(x==60)
      return 0.866;
    else if(x==90)
      return 1;
    else
      return -1;
  }
  public static double cos(int x){
    if(x==0)
      return 1;
    else if(x==30)
      return 0.866;
    else if(x==60)
      return 0.5;
    else if(x==90)
      return 0;
    else
      return -1;
  }
  public static double tan(int x){
    if(x==0)
```

```
    return 0;
  else if(x==30)
    return 0.5773;
  else if(x==60)
    return 1.732;
  else if(x==90)
    return -2;
  else
    return -1;
}
public static double cot(int x){
  if(x==0)
    return -2;
  else if(x==30)
    return 1.732;
  else if(x==60)
    return 0.5773;
  else if(x==90)
    return 0;
  else
    return -1;
}
public static double cosec(int x){
  if(x==0)
    return -2;
  else if(x==30)
    return 2;
  else if(x==60)
    return 1.1547;
  else if(x==90)
    return 1;
  else
    return -1;
}
public static double sec(int x){
  if(x==0)
    return 1;
  else if(x==30)
    return 1/0.866;
  else if(x==60)
    return 2;
  else if(x==90)
    return -2;
  else
```

```
      return -1;
  }
}
```

Main class outside package

```
import myPack.Trigonometry;
import java.util.*;
class Main{
  public static void main(String[] args){
    Scanner sc=new Scanner(System.in);
    int choice, x;
    double result;
    do{
      System.out.print("1. Sine\n2. Cosine\n3. Tangent\n4. Cotangent\n5.
Cosecant\n6. Secant\n7. Exit\nEnter a choice: ");
      choice=sc.nextInt();
      switch(choice){
        case 1:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.sin(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 2:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.cos(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 3:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.tan(x);
          if(result==-1)
```

```
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 4:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.cot(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 5:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.cosec(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 6:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Trigonometry.sec(x);
          if(result==-1)
            System.out.println("Enter from (0,30,60,90).");
          if(result==-2)
            System.out.println("Value undefined.");
          else
            System.out.println("Value is: "+result);
          break;
        case 7:
          System.exit(1);
        default:
          System.out.println("Enter a valid choice. ");
      }
```

```
    }while(choice!=7);
  }
}
```

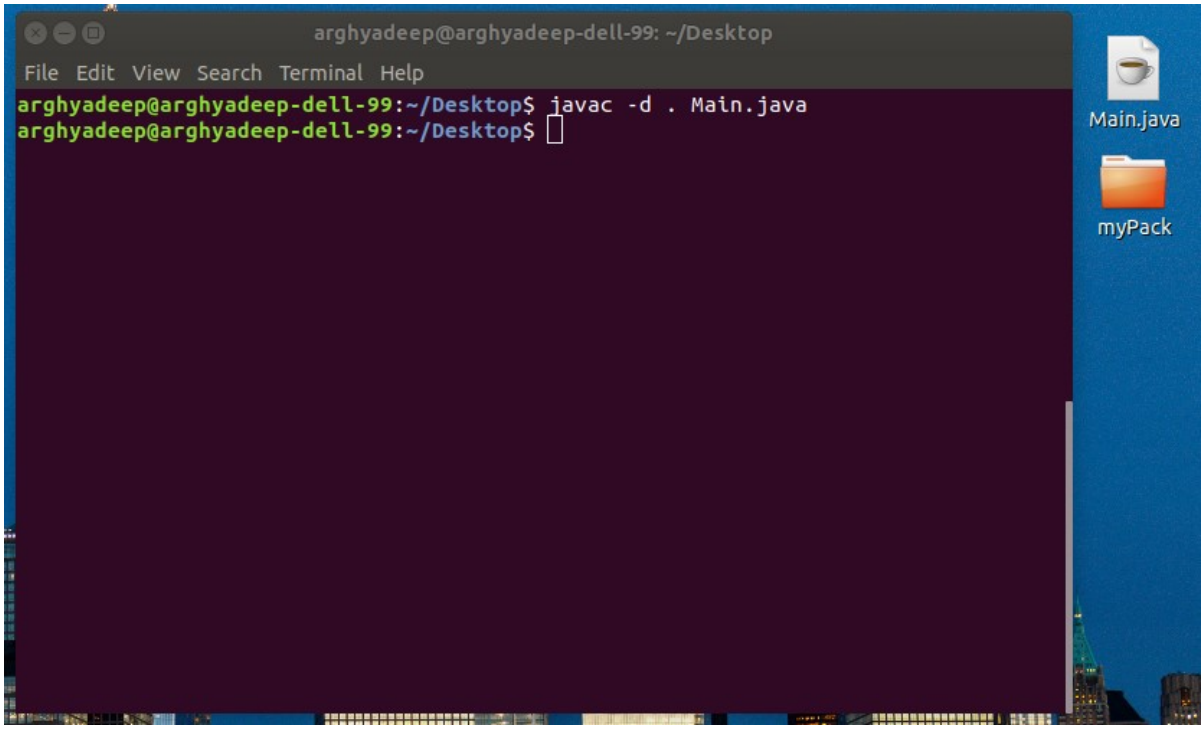
Program 3: Using built-in functions of java.lang.Math class

```
import java.util.*;
import java.lang.*;
class Main{
  public static void main(String[] args){
    Scanner sc=new Scanner(System.in);
    int choice, x;
    double result;
    do{
      System.out.print("1. Sine\n2. Cosine\n3. Tangent\n4. Cotangent\n5.
Cosecant\n6. Secant\n7. Exit\nEnter a choice: ");
      choice=sc.nextInt();
      switch(choice){
        case 1:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Math.sin(Math.toRadians(x));
          System.out.println("Value is: "+result);
          break;
        case 2:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Math.cos(Math.toRadians(x));
          System.out.println("Value is: "+result);
          break;
        case 3:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=Math.tan(Math.toRadians(x));
          System.out.println("Value is: "+result);
          break;
        case 4:
          System.out.print("\nEnter value of x(0,30,60,90): ");
          x=sc.nextInt();
          result=1.0/Math.tan(Math.toRadians(x));
          System.out.println("Value is: "+result);
          break;
        case 5:
```

```
            System.out.print("\nEnter value of x(0,30,60,90): ");
            x=sc.nextInt();
            result=1.0/Math.sin(Math.toRadians(x));
            System.out.println("Value is: "+result);
            break;
          case 6:
            System.out.print("\nEnter value of x(0,30,60,90): ");
            x=sc.nextInt();
            result=1.0/Math.cos(Math.toRadians(x));
            System.out.println("Value is: "+result);
            break;
          case 7:
            System.out.print("\nEnter value of x(0,30,60,90): ");
            System.exit(1);
          default:
            System.out.println("Enter a valid choice. ");
        }
      }while(choice!=7);
    }
}
```

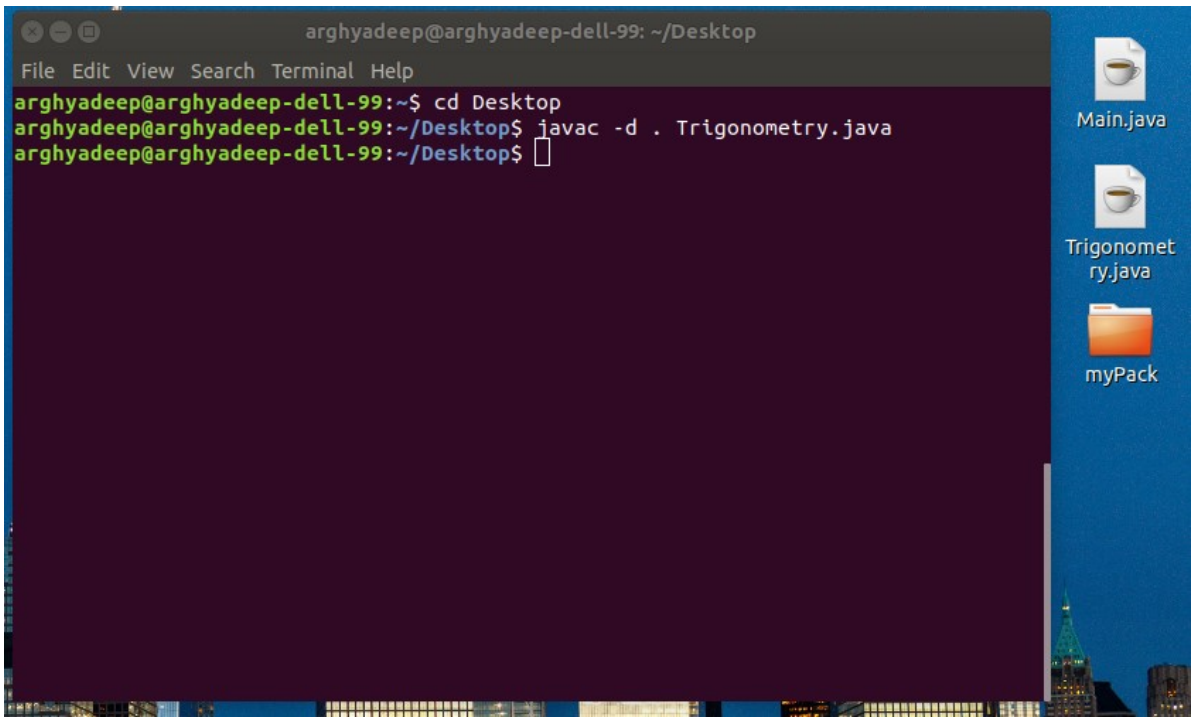
Output Screens:

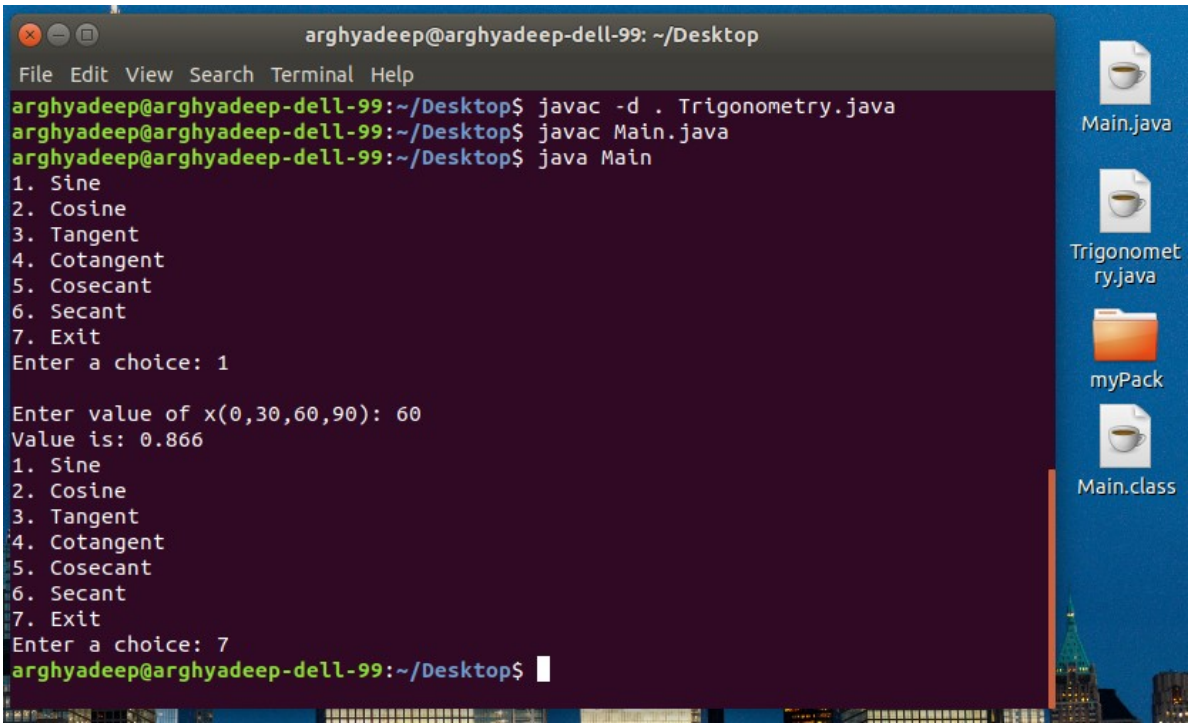
Program 1:



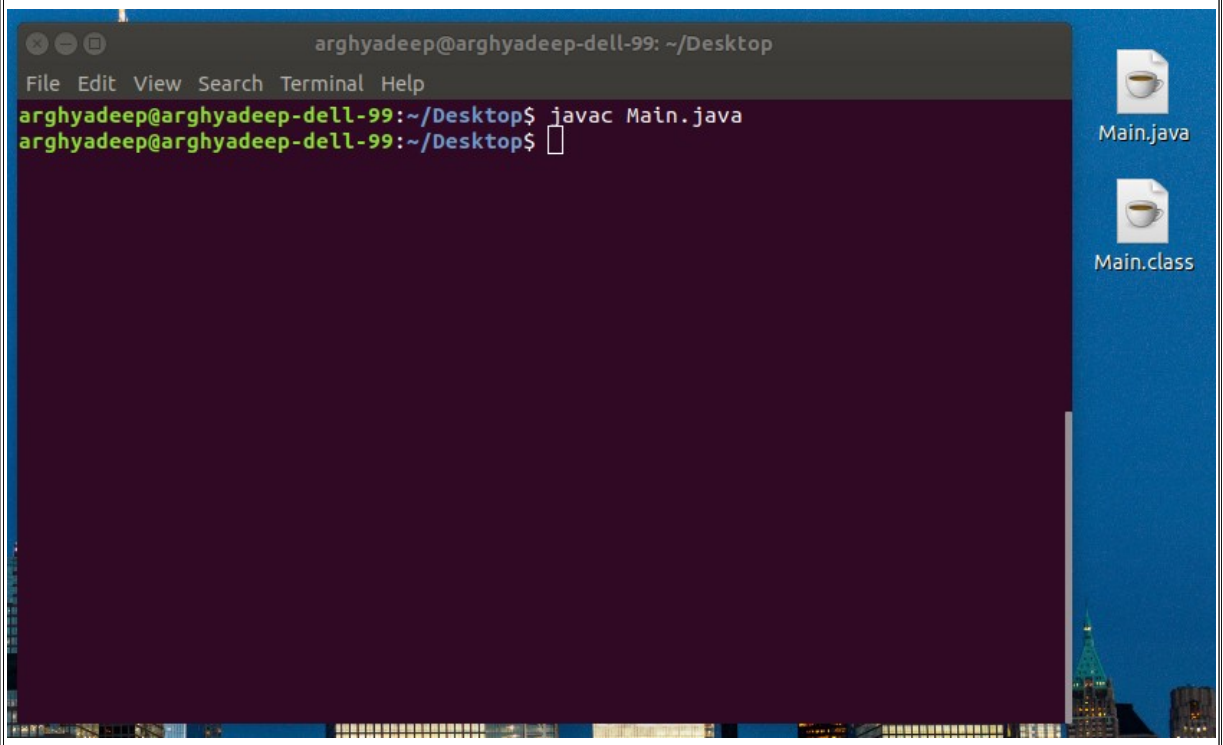
```
arghyadeep@arghyadeep-dell-99: ~/Desktop
File  Edit  View  Search  Terminal  Help
arghyadeep@arghyadeep-dell-99:~/Desktop$ java myPack.Main
1. Sine
2. Cosine
3. Tangent
4. Cotangent
5. Cosecant
6. Secant
7. Exit
Enter a choice: 1

Enter value of x(0,30,60,90): 60
Value is: 0.866
1. Sine
2. Cosine
3. Tangent
4. Cotangent
5. Cosecant
6. Secant
7. Exit
Enter a choice: 6

Enter value of x(0,30,60,90): 90
Value undefined.
1. Sine
2. Cosine
3. Tangent
4. Cotangent
5. Cosecant
6. Secant
7. Exit
Enter a choice: 7
arghyadeep@arghyadeep-dell-99:~/Desktop$
```

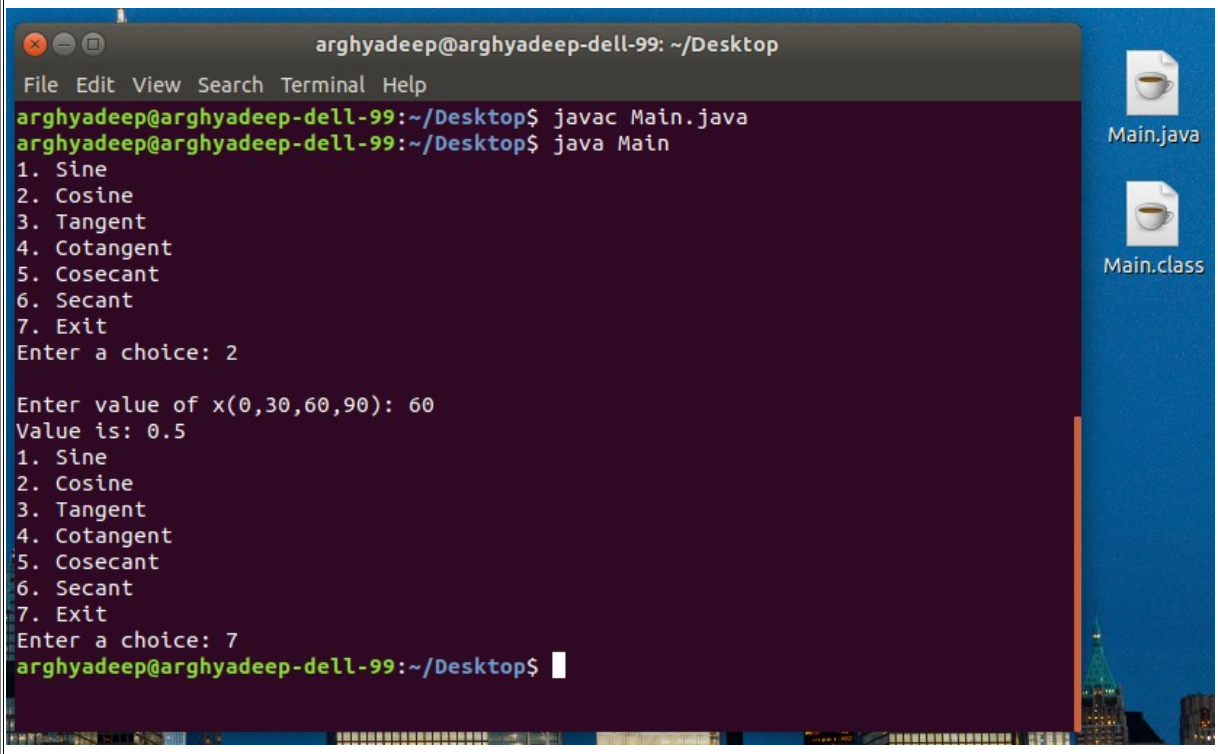
Program 2:




arghyadeep@arghyadeep-dell-99: ~/Desktop
File Edit View Search Terminal Help
arghyadeep@arghyadeep-dell-99:~/Desktop$ javac -d . Trigonometry.java
arghyadeep@arghyadeep-dell-99:~/Desktop$ javac Main.java
arghyadeep@arghyadeep-dell-99:~/Desktop$ java Main
1. Sine
2. Cosine
3. Tangent
4. Cotangent
5. Cosecant
6. Secant
7. Exit
Enter a choice: 1
Enter value of x(0,30,60,90): 60
Value is: 0.866
1. Sine
2. Cosine
3. Tangent
4. Cotangent
5. Cosecant
6. Secant
7. Exit
Enter a choice: 7
arghyadeep@arghyadeep-dell-99:~/Desktop$
Main.java
Trigonometry.java
myPack
Main.class

Program 3:


arghyadeep@arghyadeep-dell-99: ~/Desktop
File Edit View Search Terminal Help
arghyadeep@arghyadeep-dell-99:~/Desktop$ javac Main.java
arghyadeep@arghyadeep-dell-99:~/Desktop$
Main.java
Main.class



Conclusion: The programs ran successfully as we were able to use the package methods both from within the package and outside the package and got acquainted with Math class methods of java.lang sub-package.

Date: 21/09/2018 **Signature of faculty in-charge**

Post Lab Descriptive Questions (Add questions from examination point view)

1. Discuss the various levels of access protection available for packages and their implications.

2. Which package is used for pattern matching with regular expressions?

1. Java provides a number of access modifiers to set access levels for classes, variables, methods, and constructors. The four access levels are −

- Visible to the package, the default. No modifiers are needed.
- Visible to the class only (private).
- Visible to the world (public).
- Visible to the package and all subclasses (protected).

a) Default access modifier means we do not explicitly declare an access modifier for a class, field, method, etc. A variable or method declared without any access control modifier is available to any other class in the same package. The fields in an interface are implicitly public static final and the methods in an interface are by default public.

b) Methods, variables, and constructors that are declared private can only be accessed within the declared class itself. Private access modifier is the most restrictive access level. Class and interfaces cannot be private. Variables that are declared private can be accessed outside the class, if public getter methods are present in the class. Using the private modifier is the main way that an object encapsulates itself and hides data from the outside world.

c) A class, method, constructor, interface, etc. declared public can be accessed from any other class. Therefore, fields, methods, blocks declared inside a public class can be accessed from any class belonging to the Java Universe. However, if the public class we are trying to access is in a different package, then the public class still needs to be imported. Because of class inheritance, all public methods and variables of a class are inherited by its subclasses. The main() method of an application has to be public.

Otherwise, it could not be called by a Java interpreter (such as java) to run the class.

d) Variables, methods, and constructors, which are declared protected in a superclass can be accessed only by the subclasses in other package or any class within the package of the protected members' class. The protected access modifier cannot be applied to class and interfaces. Methods, fields can be declared protected, however methods and fields in a interface cannot be declared protected. Protected access gives the subclass a chance to use the helper method or variable, while preventing a nonrelated class from trying to use it.

The following rules for inherited methods are enforced −

- Methods declared public in a superclass also must be public in all subclasses.
- Methods declared protected in a superclass must either be protected or public in subclasses; they cannot be private.
- Methods declared private are not inherited at all, so there is no rule for them.

2. The package **java.util.regex** is used for pattern matching with regular expressions.