Groovy Assignments

1. **Write a groovy program to display 'Welcome To Groovy ' on command prompt.**

2. **Write a program to display Symmetrical Rangoli pattern as shown on the command line using loop control statements**

```
*                    *******
**                   ******
***        and       *****
****                 ****
*****                ***
******               **
*******              *
```

3. **Write a groovy program to read command line arguments and display them on**
   the command line.
4. **Write a groovy program to read two integer numbers from command line Strings. (Use 'parseInt' method from 'Integer' class to convert the Command string to number) , add the two numbers display the result on the Command line.**

5. Define a program similar to in assignment 2 above which will display user specified characters from command line in the Rangolis instead of '*' character.i.e.

```
C C C C C C
 C C C C C
  C C C C
   C C C
    C C
     C
```

and

```
     ?
    ? ?
   ? ? ?
  ? ? ? ?
 ? ? ? ? ?
? ? ? ? ? ?
```

6. ***Define a groovy class named 'myClass' and having following variables***
   and methods

Variables:
static Name string type
integer Num
Constructor:
Define the constructor to initialize Num and Name variables to specified values.
Static methods
setName (String newName) to assign newName to Name variable and print

the new Name value.
Instance method
getName to return the value of Name variable and print the "GetName" message.
Define the test program to create the instances of 'myClass' and assign different values to its properties and print them.

7. Define a class Bank with static int intrestRate variable and static method as follows..

```
Public static PrintMessage ( string s)
{
   System.out.println(s);
}
```

In main method create two different objects of Bank.Set different values to static variables to these objects and print the result before and after each setting of static intrestRate.
Invoke the PrintMessage method by using Class name.

8. Define a Class with methods such as following

```
Public void changeNumber (int no)
{
   no += 100;
}
public void changeString (String s)
{
    s.concat ("He");
}
public void swap (String s1, String s2)
{
   String temp=s1;
   s1=s2;
   s2=temp;
}
```

in the **main ( )** method define *int* and *string* local variables and call the above methods by passing respective parameters. Display the result / values after each methods on the console.
This will demonstrate parameter passing to the methods their effect outside the methods.
What will be required in swap method such that the changes made inside method will be reflected outside it.

9. Define a program to demonstrate type casting byte, short, long and int parameters to each other use implicit and explicit casting provide type casting to double and float parameters by giving higher and smaller range values and print the values before and after casting.

10. Define a class to print a message when the class objects are destroyed by the garbage-collector. (i.e. Provide **finalize** ( ) method.) In **main** ( ) create objects and call garbage collector to delete the objects.(make object = null before calling GC)

11. Define a **Account** class with parameters as int no, string type double balance Define different overloaded constructors and call the other constructors in overloaded constructors. Print the messages in respective constructors.
12. Define a class 'Shape' with different methods with same name 'DrawShape' but with different set of parameters. Define these overloaded methods and demonstrate their use in **main** ( ) method. Use method overloading.

13. Define a '**Base'** class with variables int acc_no, string Br_name;
    Define the method as

```
Public int SetNumber (int no)
{
   this.acc_no=no;
}
Public void SetName ( String s )
{
   Br_name=s;
}
```

    define a class '**Child'** inheriting 'Base' class with variables such as int acc_no and method as

```
Public int getAccountNumber ( )
{
     int t= acc_no;
     return t;
}
```

    in main( ) method of child class create objects of base and child classes and assign values to these objects call the respective methods using these objects and print their messages And values.

14. Modify the above classes to define constructor and print the messages inside the constructors. Inside main( ) method of '**Test'** class create different objects of 'Child' and 'Base' classes by calling specific overloaded constructors and print their values.In side the constructors try to call other constructors of the same class.
    Define 'Test Classes ' to create a base class type and assign the child object type to it.
    i.e. Child c = new child ( );
    Base b = c;
    Call the respective methods on 'b' and 'c' and print their respective variables values and demonstrate polymorphism.