1. The values passed to a method is called are ?
   1. Formal parameters
   2. Actual parameters
   3. Primitive values
   4. Objects
2. Class Square {

public int square(int x){

return x\*x;

}

}

Write a code to call this method in Square class.

1. Method overloading refers to ?
   1. A method with 10 or more parameters?
   2. A method with a very large body.
   3. More than one method with the same name but different types of parameters.
   4. More than one method with the same name with same types of parameters.
2. **public class** Program1 {  
     
    **static double** calc(**double** a, **double** b){  
     
    **double** num = a \*b;  
    b=a;  
    num = a+b;  
     
    **return** num;  
    }  
    **public static void** main(String[] args) {  
    **double** x=5.1,y =6.2;  
     
    System.***out***.println(*calc*(x,y));  
    }  
   }

What is the result of the above program ?

1. 10.2
2. compilation issue
3. 31.6
4. 0
5. **int** doIt(**int** x){  
      
    x\*x;  
   }

What is missing in the above program? Write the code to fix it.

1. The keyword void is placed in front of a method to indicate the that:
   1. The method does not return a value
   2. The method is a constructor
   3. The method is overloaded
   4. The method should return a value of an unknown type.
2. What are the different parts of a method ?
3. Write a method that takes in a int as a parmeter and perform a cube of the int and return the cube value ?
4. Write a method called maxofTwo that two integer parameters and returns the larger of two.
5. Write a method **performConcat** that takes in two string paramters and perform the concat operation of the String and return the joined string.
6. Create a class named Bank with name, id,as its members.
   1. Create a constructor with name, id as its parameters.
   2. Create an overloading constructor with id.
   3. Create an overloading constructor with name.
7. Create the accessor and mutator methods for the Bank class.
8. What is the instance data for the Bank class?