**Program 1**

// Create class A which implements the interface Number

class A implements Number {

// Implement the findSqr method to return the square of the number

public int findSqr(int i) {

return i \* i;

}

}

**Program 2**

// Create class B which implements the interface GCD

class B implements GCD {

// Implement the findGCD method to compute the GCD recursively

public int findGCD(int n1, int n2) {

// Check if either number is negative, and return -1

if (n1 < 0 || n2 < 0) {

return -1;

}

// Base case for recursion: If n2 becomes 0, return n1 as the GCD

if (n2 == 0) {

return n1;

}

// Recursive case: Find the GCD using the Euclidean algorithm

return findGCD(n2, n1 % n2);

}

}

**Program 3**

// Override the run method to specify what the thread should do when it runs

public void run() {

System.out.println("Thread is Running.");

}

**Program 4**

synchronized void print(int n){

for(int i=1;i<=5;i++){

System.out.println(n\*i);

try{

Thread.sleep(400);

}catch(Exception e){

System.out.println(e);

}

}

}

**Program 5**

// Set a new name for the thread

t.setName("NPTEL");

t.start();

1. C
2. B
3. C
4. B
5. D
6. A
7. C
8. B
9. A
10. B