1. 다음 조건을 만족하도록 스레드 PrimeNumber01 프로그램을 작성하여 클래스 Ex04의 main() 메소드에서 테스트하시오.

* 클래스 PrimeNumber01 은 1에서 지정된 생성자의 인자까지 소수를 구하여 출력하는 프로그램
* 클래스 Ex04의 main() 메소드에서 다음을 실행하면 1에서 20사이의 소수인 다음이 출력

PrimeNumber01 pn = **new** PrimeNumber01(20);

pn.start();

2 3 5 7 11 13 17 19

1. 위에서 작성한 스레드 PrimeNumber01을 참고하여 다음 조건을 만족하도록 스레드 PrimeNumber02 프로그램을 작성하여 클래스 Ex05의 main() 메소드에서 테스트하시오.

* 클래스 PrimeNumber02 은 1에서 지정된 생성자의 인자까지 소수를 구하여, 지정된 스레드 이름을 출력한 후 소수를 모두 출력하는 프로그램
* 클래스 Ex05의 main() 메소드에서 다음을 실행하면 지정된 스레드 이름이 출력되고, 1에서 20사이의 소수가 다음과 같이 출력

PrimeNumber02 pn = **new** PrimeNumber02(20, "소수 만드는 스레드");

pn.start();

소수 만드는 스레드:

2 3 5 7 11 13 17 19

1. 위에서 작성한 스레드 PrimeNumber02을 참고하여 다음 조건을 만족하도록 스레드 PrimeNumber03 프로그램을 작성하여 클래스 Ex06의 main() 메소드에서 테스트하시오.

* 클래스 PrimeNumber03은 인터페이스 Runnable을 구현하는 클래스로 1에서 지정된 생성자의 인자까지 소수를 구하여, 스레드 이름을 출력한 후 소수를 모두 출력하는 프로그램
* 클래스 Ex06의 main() 메소드에서 다음을 실행하면 지정된 스레드 이름이 출력되고, 1에서 20사이의 소수가 다음과 같이 출력

**new** Thread(**new** PrimeNumber03(25), "소수 만드는 스레드").start();

소수 만드는 스레드:

2 3 5 7 11 13 17 19 23

1. 클래스 PrintTime은 인터페이스 Runnable을 구현하는 클래스로 지정된 생성자의 횟수만큼 현재 시간과 스레드 이름을 출력하는 프로그램으로 다음 조건을 만족하도록 스레드 PrintTime프로그램을 작성하여 클래스 Ex07의 main() 메소드에서 테스트하시오.

* 클래스 Ex07의 main() 메소드에서 다음을 실행하면 다음과 같이 출력

PrintTime p1 = **new** PrintTime(3);

Thread th1 = **new** Thread(p1, "안녕하세요!");

th1.setPriority(Thread.*MAX\_PRIORITY*-1);

th1.start();

PrintTime p2 = **new** PrintTime(6);

Thread th2 = **new** Thread(p2, "반갑습니다.");

th2.setPriority(Thread.*NORM\_PRIORITY*-1);

th2.start();

순위: 4 Sat Jul 07 20:28:56 KST 2012 반갑습니다.

순위: 9 Sat Jul 07 20:28:56 KST 2012 안녕하세요!

순위: 9 Sat Jul 07 20:28:57 KST 2012 안녕하세요!

순위: 4 Sat Jul 07 20:28:57 KST 2012 반갑습니다.

순위: 4 Sat Jul 07 20:28:57 KST 2012 반갑습니다.

순위: 9 Sat Jul 07 20:28:57 KST 2012 안녕하세요!

순위: 4 Sat Jul 07 20:28:58 KST 2012 반갑습니다.

순위: 4 Sat Jul 07 20:28:58 KST 2012 반갑습니다.

순위: 4 Sat Jul 07 20:28:59 KST 2012 반갑습니다.