Multi-threaded program

→ printNumber
→ printCharacters
→ addition

(1) Create different class for different tasks
(2) extend "Thread" in each class
(3) run the tasks by using start

Notes:

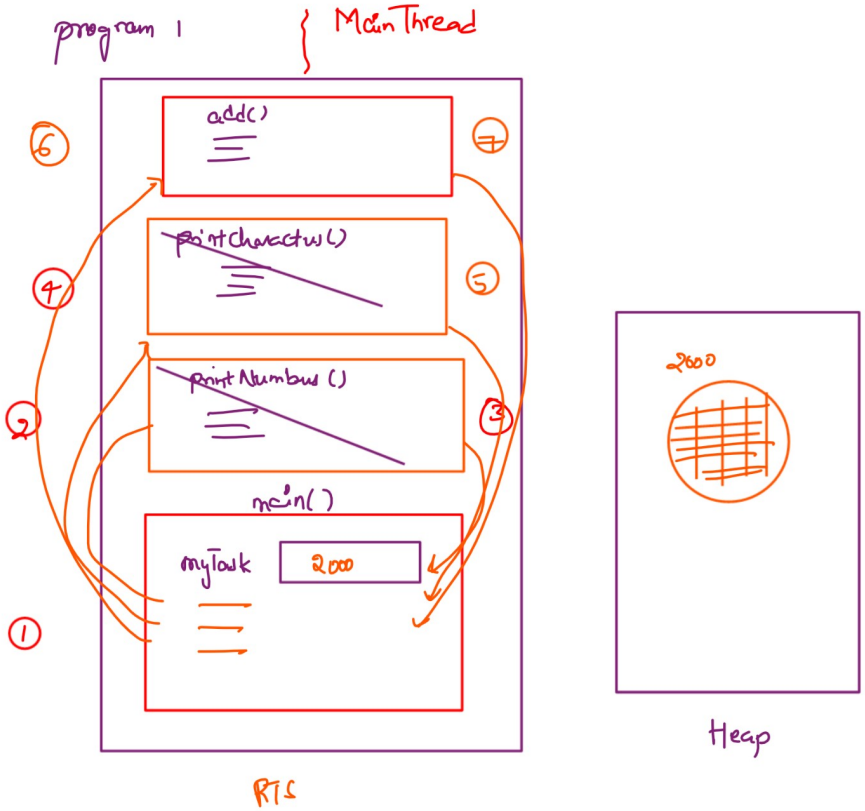
→ In prog(1), we always get a regular output because it is single threaded program. In other words only one thread executes all the different tasks

→ In prog(2), we will get different outputs every time since it is multithreaded program, In other words, there are 4 threads which is executing that program & all the 4 threads are independent each other

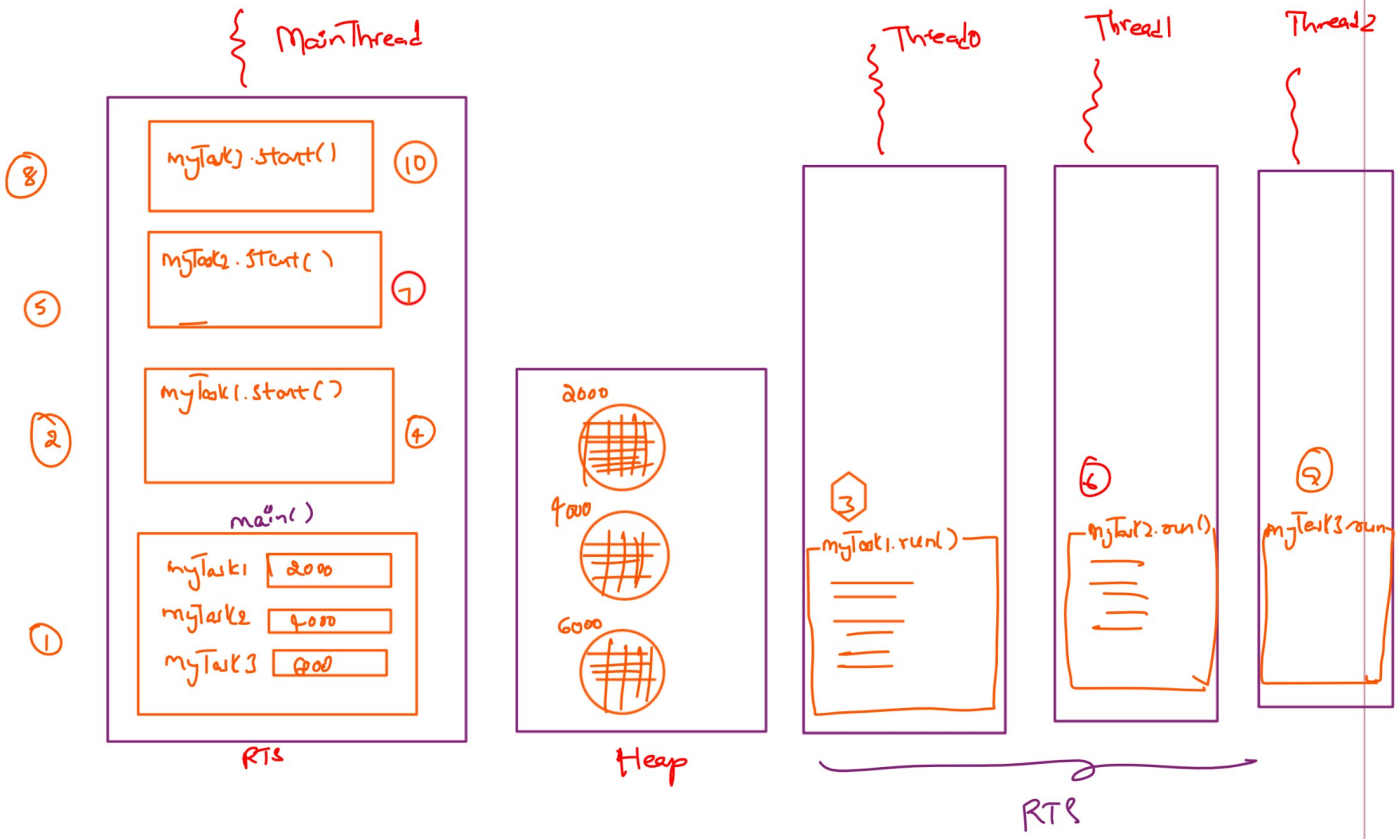
4 = (3) + (1)

(3) → Child thread
(1) → Main thread

Memory Map : program 1



Memory mapping : program 2



Steps involved by the start() :

→ Registers the thread

→ Creat the new stack

→ Call the run() & pushes the stack frame of the run() to the newly created stack

Notes :

→ We can execute the above program(2) by directly calling the run method as well, but it will not lead to multithreaded program, because no new thread nor stack will be created, when we call the run() directly

→ by the help of start(), it will register a new thread & creates a new stack, which leads to multithreaded program