Day 12 - 104608492 - Shirisha Perapagu

Task 1

Data Structures

Data Structures are different ways to store or organize our data or large amounts of data so it easy to fetch and use whenever required efficiently.

Example:

Bank stores its customer data in large amounts, bank needs to organize these data in some format so it can use List DS etc.

Task 2

Types of Data Structures:

1.List/ArrayList

2.LinkedList

3.Stack

4.Queue

5.Set

Task 3

Operations in Data Structures

|  |  |
| --- | --- |
| DS Operation | Meaning |
| Insert or Add | Adds a new item into DS |
| Delete or Remove | Take out an item |
| Search / Find | Look for a specific item |
| Update | Change the value of an existing item |
| Sort | Arrange items in order (ascending or descending) |
| Push | Add to the top of Stack |
| Pop | Remove from top of Stack |
| Peek / Top | Look at the top of Stack without removing |
| Traverse | Go through each item one by one |
| Length | Count how many items in DS |

Task 4

What are static and dynamic arrays? Explain or summarize key points in a table like Size, performance, memory, flexibility, limitations

|  |  |  |
| --- | --- | --- |
| Feature | Static Array | Dynamic Array |
| Size | Fixed after creation | Can grow or shrink as needed |
| Performance | Fast as it has direct memory access | Slower due to resizing |
| Memory Usage | Pre-allocated | May need extra memory based on resizing space |
| Flexibility | Not Flexible as size must be known early | Very flexible it adjusts as needed |
| Limitations | Memory wastage in case size is too big | Uses more memory temporarily during resizing |

Task 5

What is the binary value of a?

Decimal value of a is 97

Converting it into binary

01100001

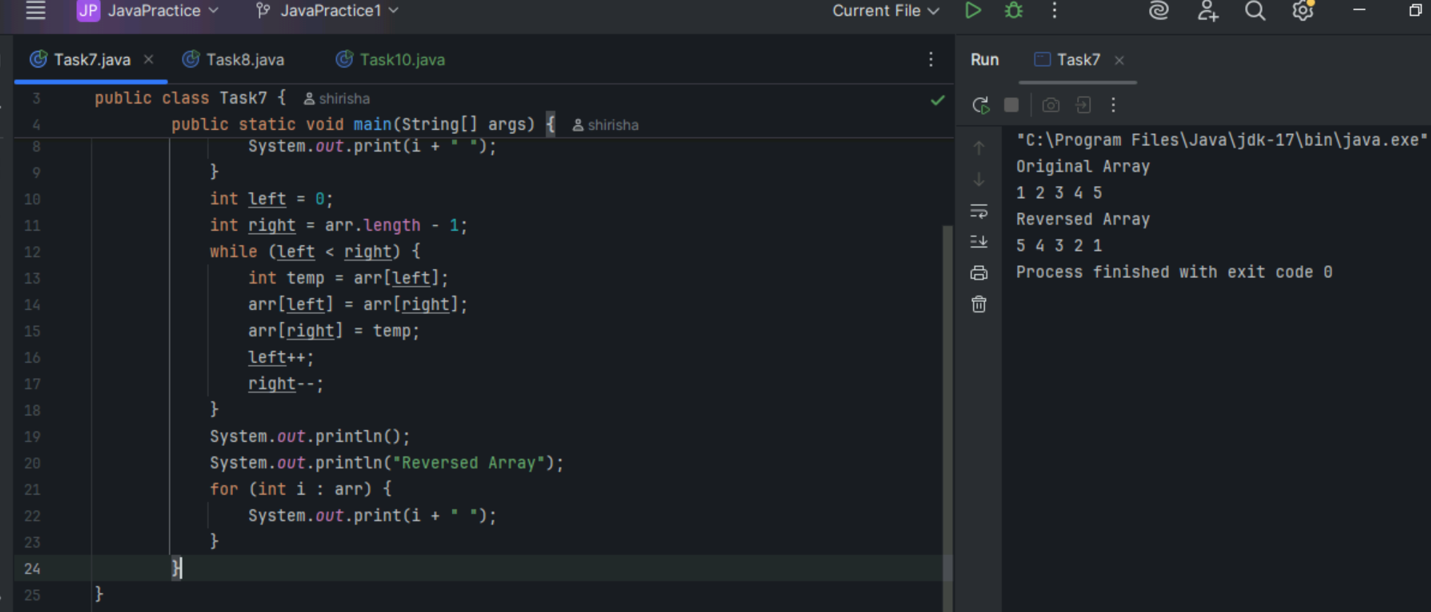
Task 6

Types of Computer Memory

|  |  |
| --- | --- |
| Memory Type | Meaning |
| RAM – Main memory | Temporary memory used when computer is running |
| ROM | Permanent memory that stores important startup instructions(already written) |
| Cache | Very fast memory that stores recently used data for quick access. |
| DRAM – Dynamic RAM | Slower and cheaper memory used as main RAM that needs constant refreshing |
| SRAM – Static RAM | Faster and expensive memory used inside CPU that doesn’t need refreshing |
| MROM – Masked ROM | Memory where instructions are permanently written and can’t be changed at all |
| PROM – Programmable ROM | Blank ROM where we write to once and then it will be permanent |
| EPROM – Erasable PROM | We can erase using UV light and rewrite with new data |
| EEPROM – Electrically Erasable PROM | We can erase and rewrite using electricity without removing it |
| Virtual Memory | Hard drive space used as extra RAM when real RAM is full |

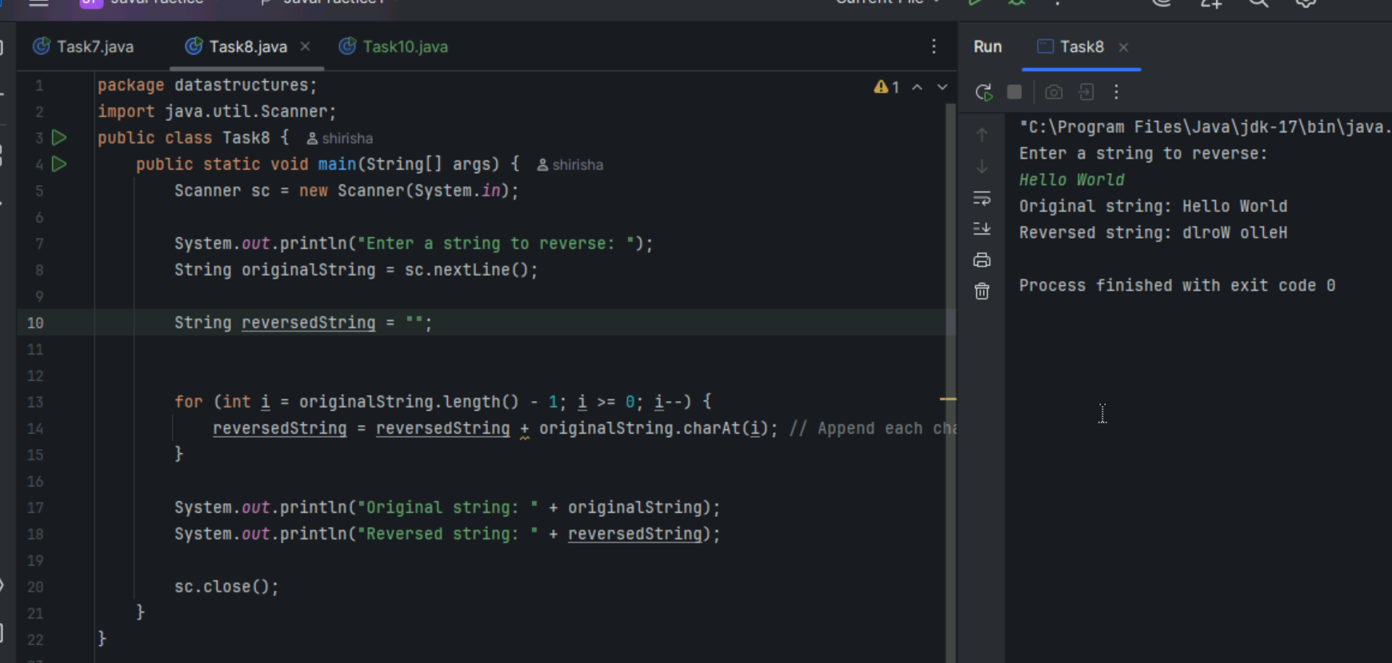
Task 7

Code to reverse an array



Task 8

Code to reverse string using user input



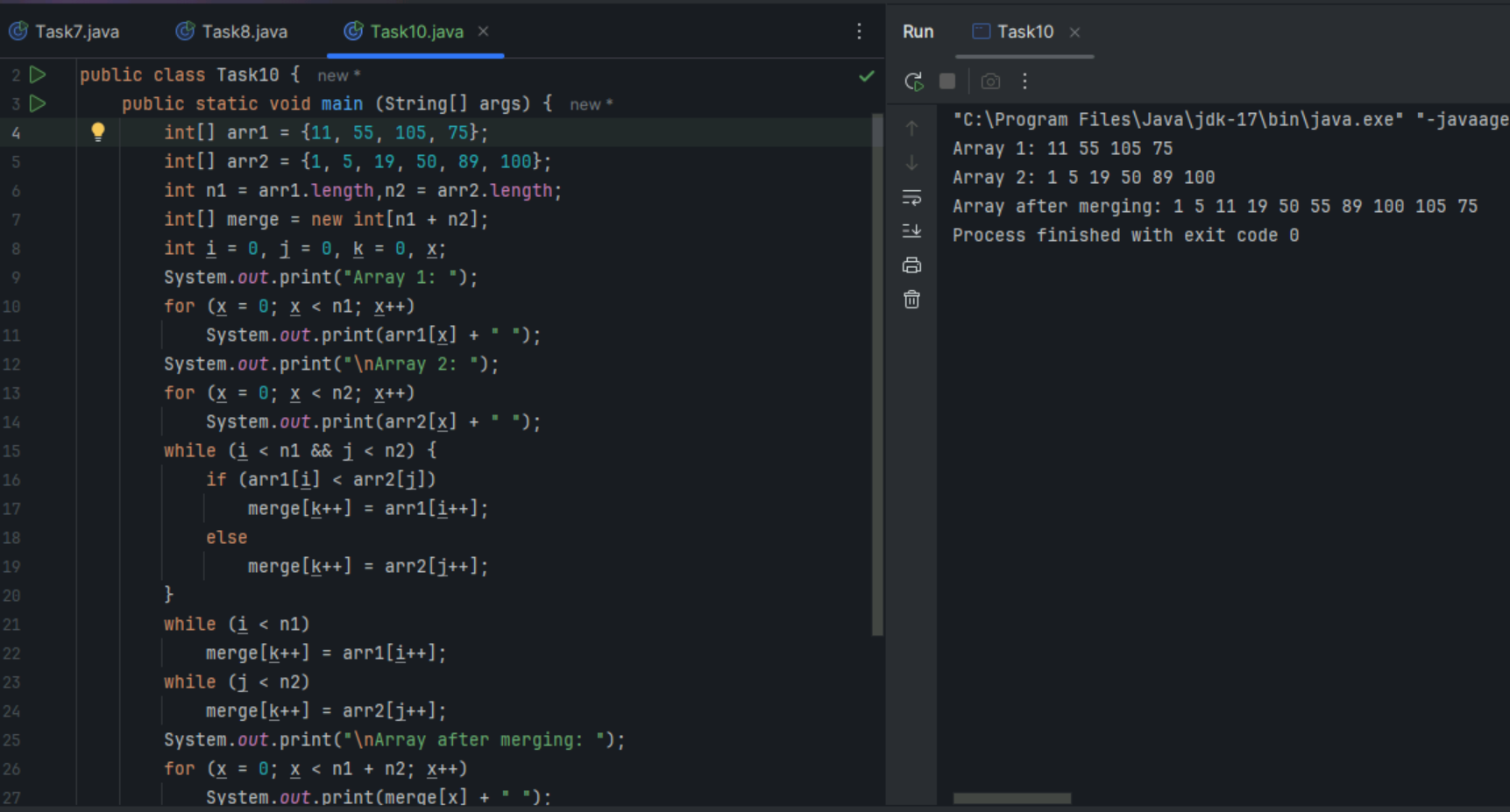
Task9

If you don’t have leetcode or hackerrank, create it.

Already have leetcode

Task 10

Merging arrays



Task 11

What do you understand by Hash Table?

Hash table is a data structure which stores data in key-value pairs.

We can access or find the data via key which is unique.

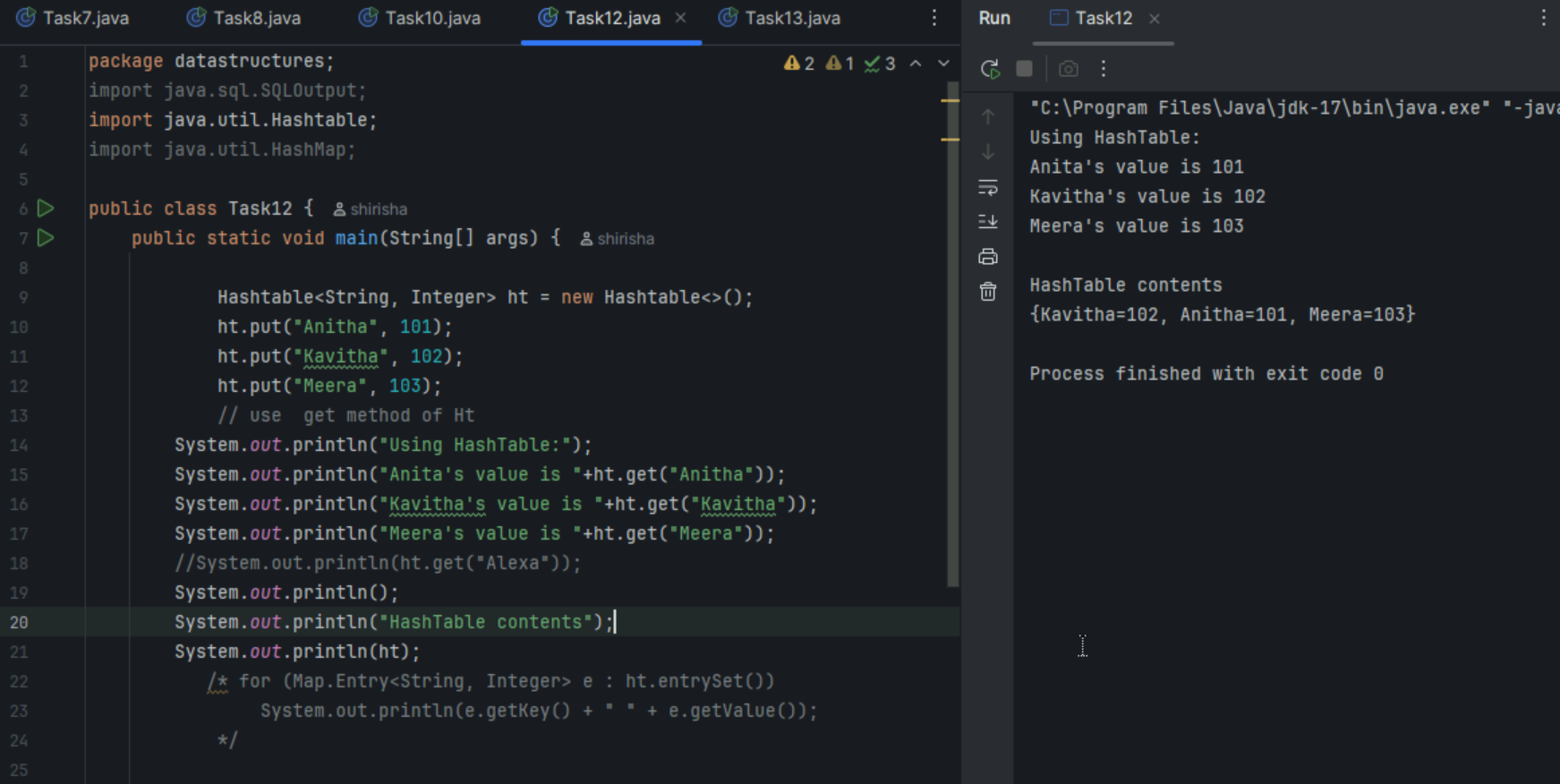
Ex: Roll no. – Student names in a College like (100 , ”Shirisha”) – (key , value)

We can access student name (value) via unique roll no. (key).

It is a faster process, we just give key and it gives us back its value.

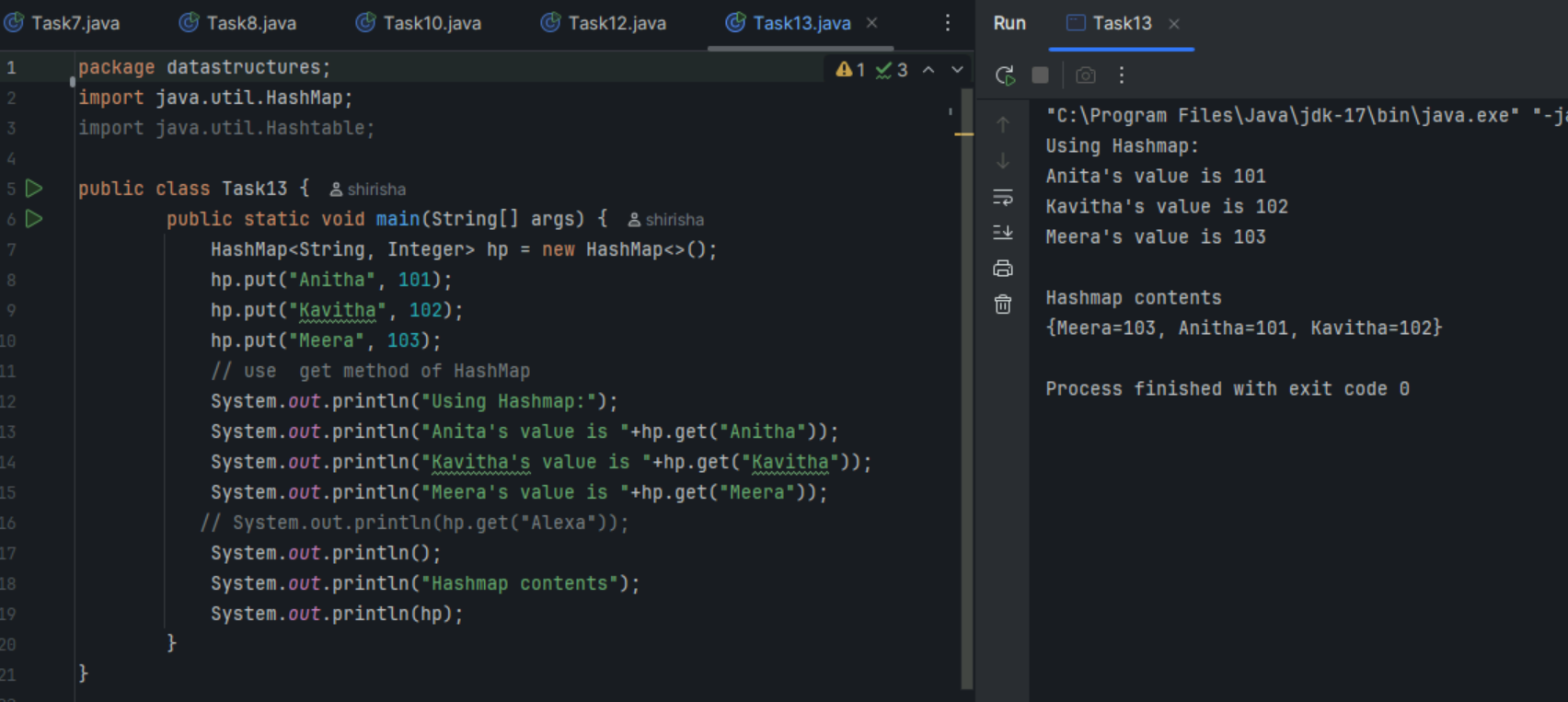
Task 12

Understand the below Hashtable code and try to print values using get method of Hashtable



Task 13

Wap to create  a HashMap and display them.



Task 14

Similarities and Differences between Hashtable and HashMap

Similarities:

|  |  |
| --- | --- |
| Feature |  |
| Key-value pair | Both store data in key-value pairs. |
| Hashing | Both use hashing techniques to store and access values. |
| Interface | Both implements Map Interface |
| Collisions | Both use linked lists for handling collisions. |

Differences:

|  |  |  |
| --- | --- | --- |
| Feature | Hashtable | HashMap |
| Thread Safety | Synchronized so it is Thread-safe | Asynchronized so not Thread-safe |
| Performance | Slower due to synchronization | Faster due to no synchronization |
| Null Keys | Not allowed | Yes, one null key allowed |
| Null Values | Not allowed | Yes, many null values allowed |
| Legacy or modern | Legacy class | Modern - part of Java Collections Framework |
| Inheritance | Extends Dictionary class | Extends AbstractMap |

Task 16

1. Adding null key value twice

2. Making HashMap synchronized

