Core Java

11. Collections

Collections

- Collections are ways to handle multiple objects with ease
- Problems arising due to a huge number of variables otherwise
- Arrays are not the only way
- Java provides collections in addition to standard arrays
- Easier to work with the Java Collections Framework than with arrays

Array

- An array in Java can be used to hold objects of the same type together
- In C, Strings were arrays of characters which terminated with the NUL character
- People got tired of having to memory manage arrays; String class holds such an array internally, does lots of boilerplate work
- Arrays are still used for performance or memory issues

Array

- The arguments passed to *public static void main* are an array
- *String []args*
- Access length of array using *.length* which is treated as a member. Not a function call!
- *int nargs=args.length;*
- Array elements can be accessed by subscript, an integer from *0* to *arr.length-1*: *arr[i]*

Syntax

```
int []arr2;
int arr[],arr3[30]={2,3,4,5};
arr=new int[20];
int [][]arr4;
arr4=new int[20][];
arr4[0]=new int[10];
arr4[1]=new int[2];
```

Collections

- Collections are usually found in *java.util* package
- Besides, some other third-party collection frameworks are available (Jakarta Commons Collections)
- Use Collections to save time, reuse code & avoid bugs
- Collections usually work with objects than primitives

Collections of Objects

- If not primitives, why use it?
- Autoboxing provides a way to case primitives to objects
- Example: *Integer a=4;*
- Those objects can now be added to Collections
- Disadvantage is memory usage
- Advantage is lesser code (no need for function overloading), better maintenance

ArrayList

- *java.util.ArrayList* is a replacement for Arrays in Java
- Is a subclass of *java.util.List*
- Other implementation of *List* is *LinkedList*
- Important methods: *length()*, *add()*, *get()*, *remove()*
- Other method of access is through Iterators
- Enhanced for-loop makes *Iterator*s easier

HashMap

- *HashMap* is a particular implementation of *java.lang.Map*
- It allows you to store unique objects and retrieve them fast
- Used for attributes, properties
- Useful methods: *get(), put(), remove()*
- *keys()* and iterate through property list

Demonstration

- Compile and Execute a few programs

Questions?