1.

A. (Employee e) -> e.getName()

Function<Employee, String> getName1 = (Employee e) -> e.getName();
Function<Employee, String> getName2 = Employee::getName;
Method reference type: Class::instanceMethod

B. (Employee e,String s) -> e.setName(s)

BiConsumer<Employee ,String > setName1 = (Employee e,String s) -> e.setName(s);
BiConsumer<Employee ,String > setName2 = Employee::setName;
Method reference type: Class::instanceMethod

C. (String s1,String s2) -> s1.compareTo(s2)

Comparator<String> compareTo1 = (String s1,String s2) -> s1.compareTo(s2);
Comparator<String> compareTo2 = String::compareTo;
Method reference type: Class::instanceMethod

D. (Integer x,Integer y) -> Math.pow(x,y)

BiFunction<Integer, Integer, Double> pow1 = (Integer x,Integer y) -> Math.pow(x,y);
BiFunction<Integer, Integer, Double> pow2 = Math::pow;
Method reference type: Class::staticMethod

E. (Apple a) -> a.getWeight()

Function<Apple , Double> getWeight1 = (Apple a) -> a.getWeight();
Function<Apple , Double> getWeight2 = Apple::getWeight;
Method reference type: Class::instanceMethod

F. (String x) -> Integer.parseInt(x);

Function<String , Integer> parseInt1 = (String x) -> Integer.parseInt(x);
Function<String , Integer> parseInt2 = Integer::parseInt;
Method reference type: Class::staticMethod

G. EmployeeNameComparator comp = new EmployeeNameComparator();
(Employee e1, Employee e2) -> comp.compare(e1,e2)

Comparator<Employee> compare1 = (Employee e1, Employee e2) -> comp.compare(e1,e2);
Comparator<Employee> compare2 = comp::compare;
Method reference type: object::instanceMethod