iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37249

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Suitability of Scala

Vaishnavi Gupta
(20MSM3071), Chandigarh University, Gharuan Mohali, Punjab- 140413, India

Abstract: ***We all are not new to the fact that data is increasing exponentially. It is increasing more than ever before, from 2 zetabytes in 2010 to 59 zetabytes in 2020. Not only this, it is predicted to be 149 zetabytes by 2024. As the data is growing and that too how the traditional methods like SQL, Excel, etc., would not be enough and feasible to analyze or store huge amount of data as they have storage and processing restrictions. It was important to use tools which are much more efficient and much more effective. To handle this 'Big Data' various tools were created E.g.: Hadoop, Spark, NoSQL, etc., not only this various programming languages also came into picture. Scala is just one of them. In this paper we are trying to highlight the importance of Scala, and how it is so unpopular and under-rated as compared to other programming languages; despite the fact that it is a strong statically typedgeneralpurpose programming language which supports both object oriented language and functional programming. Also, it is easy to understand and use as compared to other programming languages. In this paper we will see and discuss how Scala has been beneficial and why do we want it to gain more popularity in the industry. We will also see the cons associated and if it's worthy enough given its cons.***

I. INTRODUCTION

By big data we mean large volume of data which can be both structured or unstructured data.For instance, we generate tons and tons of data daily using our mobile phones in the form of text, phone calls, emails, photos, videos, searches and music. Approximately 40 exabytes of data gets generated every month buy a single user. Imagine this number multiplied by 5 billion smart phone users. This amount of data is quite a lot for the traditional computing systems to handle and this massive amount of data is what we term as big data. If we take a look at the data generated per minute on the internet, we find 2.1 million snaps are shared on snapchat, 3.8 million search queries are made under google, 1 million people log in their Facebook ,4.5 million videos are watched in you tube, 188 million emails are sent. This is a lot of data. We can classify any data into big data with the help of **5V's**namely **V**olume**, V**elocity, Variety,**V**eracity and **V**alue. To understand this, we can take the example of healthcare industry. Hospitals and clinics across the world generate volumes of data. 2,314 exabyte of data is generated annually here (volume). It is in the form of patient record and test results; all this data is generated at a very high speed (velocity). This data can be in various forms like structured, semi structured or unstructured data, for example excel records, log files and X-ray images respectively(variety). Accuracy and trustworthiness of the generated data is termed as veracity. Analyzing all this data will benefit the medical sector by enabling faster disease detection, better treatment and reduced cost. This is called the value of big data. We store and process this big data using various frameworks like Cassandra, Hadoop, Spark and Scala.

Table 1 : Traditional (RDMS) Data vs Big Data

Mostly structured data	Can be structured, semi structured or unstructured
Stored in megabyte, gigabyte, terabyte	Stored in petabyte, exabyte
Increases gradually	Increases exponentially
It is locally present, centralized	Globally present, distributed
Processed using oracle, sql server	Processed using Hadoop, spark
Single node	Multinode cluster

II. NEED OF BIG DATA ANALYTICS

We create 2.5 quintillion of data every day. Data comes in from various sources and has different formats. This enormous collection of data is big data. We need big data analytics for

1) *Making Smarter And More Efficient Organizations:* New York police department is utilizing data patterns, scientific analysis and technological tools to prevent the occurrence of crime. Big data analytics is helping many police departments to identify the criminal activities before it occurs. They analyze the entire big data technology to geolocate and analyze the historical pattern. They utilize the data patterns, technological tools, scientific analytics to their jobs and they are ensuring that that with the help of this approach, they are doing their jobs to their best ability. Using these techniques, they are able to detect the crime hotspots and there they deployed their local officers to reach there on time before the crime is committed.

2) *To Optimize Business Operations By Analyzing Customer Behavior:* The best example for this is amazon. They use click stream data and the historical purchase data of more than 300 million customers which have signed up for amazon. They analyze each user's data, how they are clicking on different products and how they are navigating through their site. So basically, they show each user customized results on customized webpages. So, after analyzing all these clicks of every user they are able to better understand their site navigation behavior, the path people are taking to buy their product and services and what else a customer looked on while buying that product and also the path that a customer led to leave their page. So, this information helps amazon improve its customer experience and hence expand their customer base
3) *Cost Reduction:* Big data technologies like Hadoop and cloud-based analytics basically reduce your cost significantly for storage of big data, because for storing big data, if you buy huge servers and machinery that costs a lot. By using Hadoop technology, data is being stored in distributed file system so it reduces the cost a lot. Healthcare is using big data analytics to curb their cost. using new data tools that send automatic alerts when patients are due for immunization or lab works, more and more doctors could reduce the hospitalization by practicing better preventive care. Patients started using these new sensor devices at home.so, these new sensor devices basically deliver constant streams of data that can be monitored and analyzed in real time. So, they help the patients avoid hospitalization by self-managing their conditions. Now, for hospitalized patients' doctors can use predictive analytics to optimize outcomes and then reduce the re-admissions. Parkland hospital uses analytics and predictive modeling to identify high risk patients and predict likely outcomes once patients are sent home. As a result, parkland reduced 30-day re-admissions for patients with heart failure, by 31 percent.
4) *Next Generation Products:* big data is contributing to generate more such high-tech products to see how customer needs can be satisfied and how they can use these new generation products for their own benefit. Some examples are-

a) Google self-driving car, it makes millions of calculations on every trip that helped the car decide when and where to turn, when to slow down or speed up or when to change the lane, so the same decision a human driver is making behind the wheel google self-driving car is also doing that with the help of big data analytics.

b) Another example is Netflix. Netflix committed for two season of itsextremely popular show "house of cards" without even seeing a single episode of the show. This project of house of cards of 2 seasons costed Netflix about 100 million dollars. Netflix was able to take such a big risk monetarily by big data analytics. Using the viewer data, the company was able to determine the fans of the original house of cards, which aired in the UK. So basically, Netflix is recording everything to what shows you are watching to when you pause it or when you turn it off. So last year Netflix grew its subscribers by around 10 % and then they added nearly 20 million subscribers from all around the globe.

c) Another amazing example is smart yoga mat. This has sensors embedded in the mat which will be able to provide feedback on your postures score your practice and even guide you through an at home practice so the first time you use your smart mat it will take you through a series of movements to calibrate your body shape, size and personal limitations. So, this personal profile information of yours is then stored into your smart mat app and this will help the smart mat detect when you are out of alignment or balance so overtime it will automatically evolve with updated data as you improve your yoga practice.

III. ELEMENTS

1) *HADOOP:* It consists of three components that are specifically designed to work on big data. In order to capitalize on data, the first step is storing it. The first component of Hadoop is storage unit, the Hadoop Distributed File System (HDFS). Storing massive data on 1 computer in infeasible hence data is distributed amongst many computers and stored in blocks. If you have 600-megabyte data to be stored, hdfs splits data into multiple blocks, that are then stored on several data nodes in the cluster. 128 megabytes is the default size of each block. Hence 600 blocks will be split into 4 blocks of 128 mb each and the last 88 mb is stored in fifth block. If one data node crashes, we do not lose the data within because hdfs makes copies of data and stores it across multiple systems. When block 1 is created, it is replicated with a replication factor of 3 and stored on different data nodes, this is the replication method. Thus,hdfs is fault tolerant.After the data is stored successfully, the second component of Hadoop, i.e., processing comes into play.Traditionally the entire data was processed in a single machine having a single processor, this consumed time and was inefficient especially when processing large volumes of a variety of data to overcome this MapReduce splits data into parts and processes each of them separately on different data nodes. The individual results are then aggregated to give the final output. Third component of Hadoop is yarn, yet another resource negotiator or yarn consists of a resource manager, node manager, application master and containers. The resource manager assigns resources. Node managers handle the nodes and monitor the resource usage in the node. The containers hold a collection of physical resources.

2) *Mongo DB:* It is a popular, document-oriented NoSQL database. It was initially released on 27 august 2009. It replaces the concept of rows of conventional relational database models with something called documents. It offers the developers the flexibility to work with evolving data models. Since it is document based mongodb allows embedded documents arrays and represents complex hierarchical relationships using a single record. It is also schema free, which means that the keys defined in the document are not fixed. As a result, massive data migrations can be ruled out. Mongodb is widely used because of the following reasons

a) *Flexibility:* Mongodb's notion of documents that can contain sub-documents nested in complex hierarchies is really expressive and flexible. A user can selectively index some part of a document or a query based on attribute values, regular expressions or ranges

b) *Native Aggregation:* Allows its users to extract and transform data from mongodb and either load them in a new format or export it from mongodb to other data sources, it makes it extremely compatible.

c) *Schema free Model:* Applications get the power and the responsibility to interpret different properties in different ways

3) *Spark:* Apache spark is an open-source distributed processing system which we use for big data. It makes use of in memory caching and optimized query execution for fast queries against any size data. It is a fast and general engine for large scale data processing. By fast we mean it is a faster approach to work with big data as compared to classical MapReduce. Spark runs fast because it runs on random access memory (RAM). We can use spark for multiple things like we can run distributed sql, we can create data pipelines, we can input data in a database, we can easily run machine learning algorithm and many more. Some very good features of apache spark are:

a) *Real time Processing:* MapReduce processes only stored data, contrary to this spark can process real time streaming data and hence it can produce instant outcomes.

b) *In Memory Computing:* As we know, the data is stored in RAM of servers with the help of which we get quick access and the speed of analytics is enhanced.

c) *Flexibility:* It supports multiple languages and it is a multi-paradigm language. Gives developers the advantage of writing applications in java, Scala, R or python

d) *Fast Processing:* apache spark took over all the big data technologies because of its speed. Big data needs to be processed at very high speed. Spark has Resilient Distributed Dataset (RDD) which saves up a lot of time, making it many times faster than Hadoop.

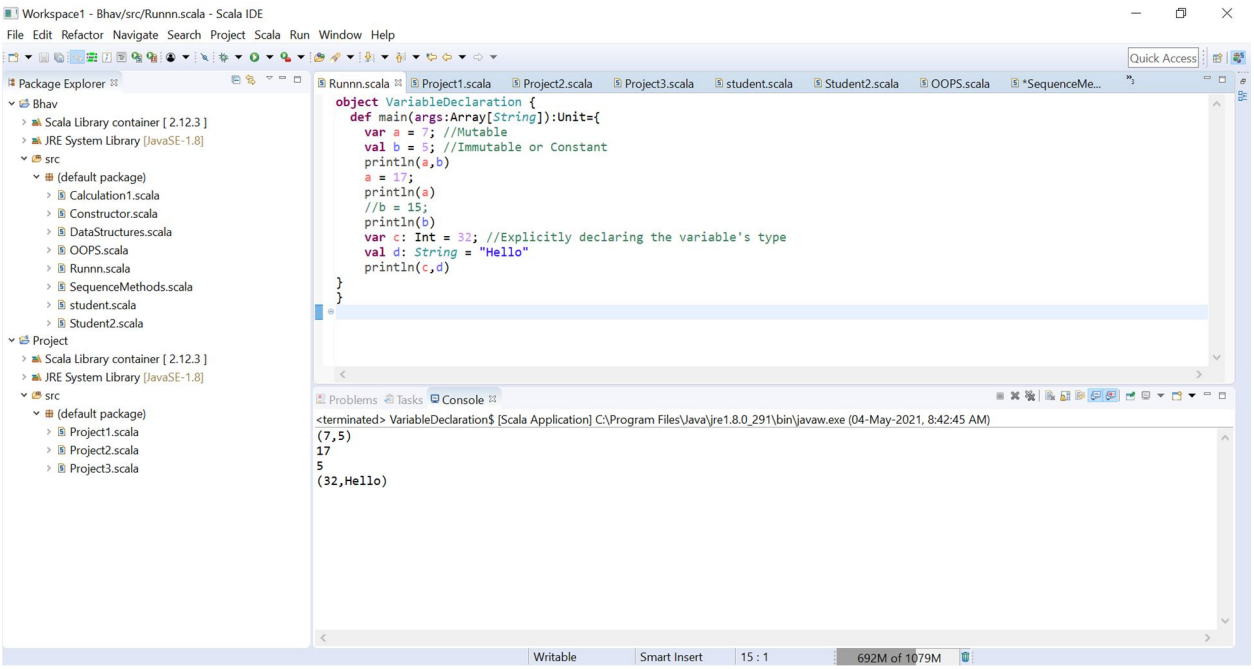
4) *Scala:* Scala is a programming language invented by Martin Odersky in 2001. The name Scala depicts about the scalability of the languageoffers. Scala minimizes the code length and reduces the execution time. It is basically derived from java. It is said to be a multi paradigm language. Paradigms are essentially a set of principles that we follow for a certain language. Set of strategies or principles followed to program a certain language or solve certain problems. It brings together object-orientedprogramming and functional programming into one concise programming language. So, object oriented, like java for example we have these objects these variables in Scala are all saved as objects. So, this is what makes it a powerful object-oriented programming tool. However, it also provides functional programming and it works in that way making it really easy for us to move the 2 together without having to combine different languages and it's all into 1 concise programming language. variables in Scalaare implicitly saved as an object by default. It provides a lightweight syntax for defining functions. Variable types are explicitly declared and known at compile time; this makes this system more complex.

Scala catches more errors at compile time, making execution faster. Dynamic typing requires more testing to ensure there are no bugs and is slower.it runs on the java platform (java virtual machine). The original goal of Scala was to address the issue of java. Scala.js is a Scala compiler that compiles to java script, making it possible to write Scala programs that can run in web browsers. It has become a staple language for data science, alongside R and python.

IV. PROGRAMMING IN SCALA

As we already discussed Scala supports both functional and object-oriented programming. We'll be now be discussing basic and yet very important concepts in Scala Programming.

We have used eclipse, though one can work on any IDE.



On the right-hand side, we can work on our projects, the console window is also attached below the working space only. While on the left-hand side hand we can see all our projects are listed.

Let's start with our very first program in Scala i.e., let's get started with variable declaration and initialization only.

We all know, variables are broadly divided in 2 types:

1) *Mutable:* Mutable variables are those in which we can update the value of our variable in the whole program.
2) *Immutable:* Immutable variables are the one in which we can't update or modify the value of our variable in our program.

In Scala, we declare Mutable variable with 'var' keyword and Immutable variable with 'val'.

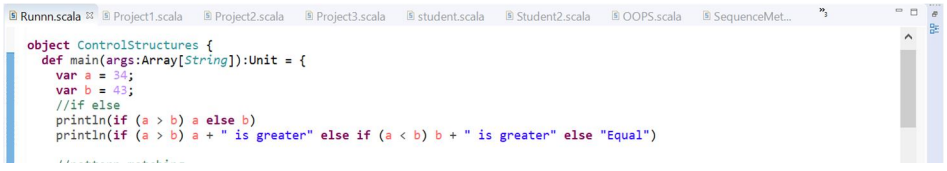
In the above example, we have declared 'a' and 'c' as Mutable ones and 'b' and 'd' as Immutable ones.

We print our results by using 'println()' function in Scala. We can see the output has been printed in the console window.

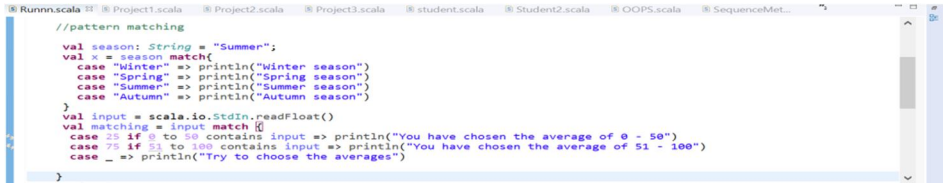
Note: In Scala there is no need to define the data type explicitly, it does that for us on its own, though we can do that it if we want.

A. *Now we'll look at very famous Control Structures*

1) *If else Statement:* Compared to other programming languages it is the easiest to implement if else statement in Scala. As we can see in the example, we used only one line of code and that's it! Our code looks so neat.



2) *Pattern matching:* Popularly known as switch case in other programming languages, this comes very handy when we need to match the patterns/trends and take decisions accordingly. We have used 2 programs in our example to illustrate how we can also take input from the user at the run time.



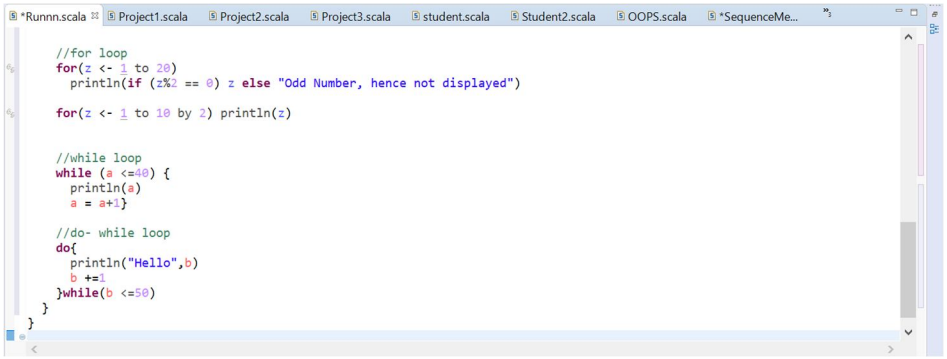
We can clearly see how the input is getting matched with the cases provided and how we are able to take decisions accordingly.

3) *Now comes 'The Loops':* Loops are used when we need to iterate over, i.e., we need to execute the statement multiple number of times for different values.

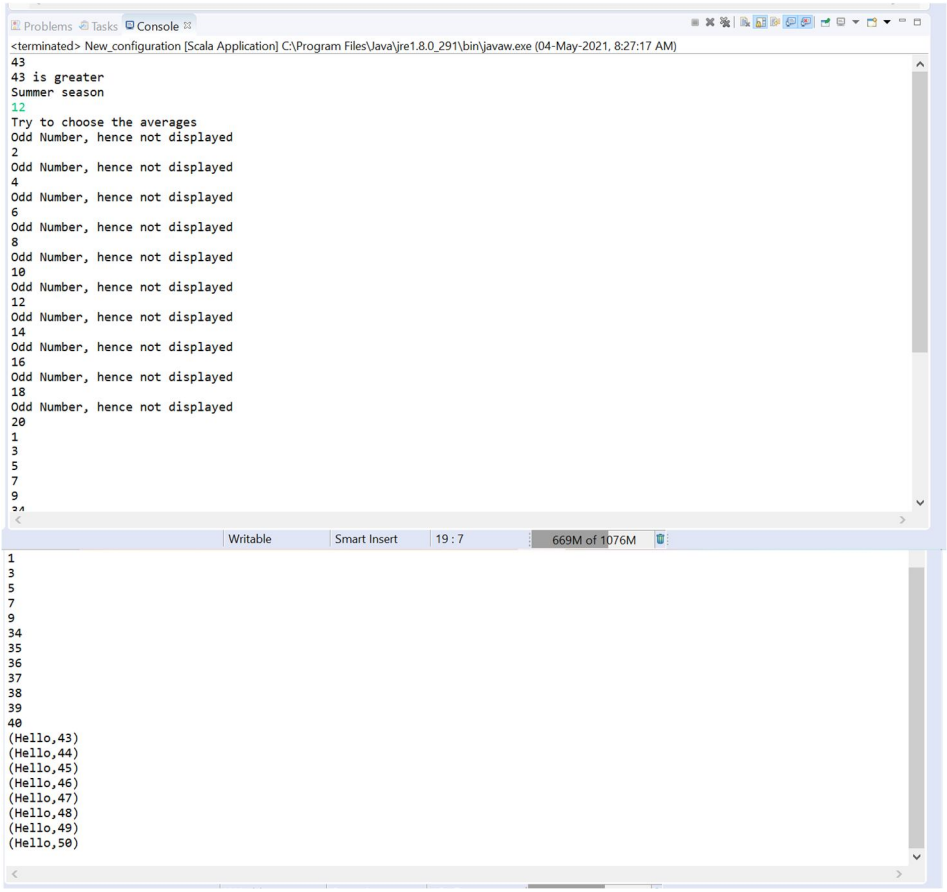
a) *For Loop:* We can also use for loop in a single line.

b) *While Loop:* While loop is kind of same only as we use in other programming languages.

c) *Do While:* Do while is also nonetheless same.

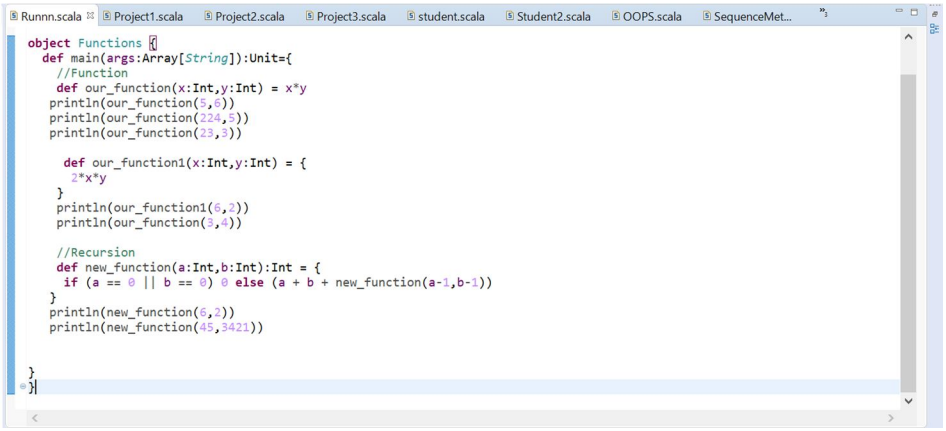


Due to the space restrictions we have shown the outputs of all the above programs in one place only.

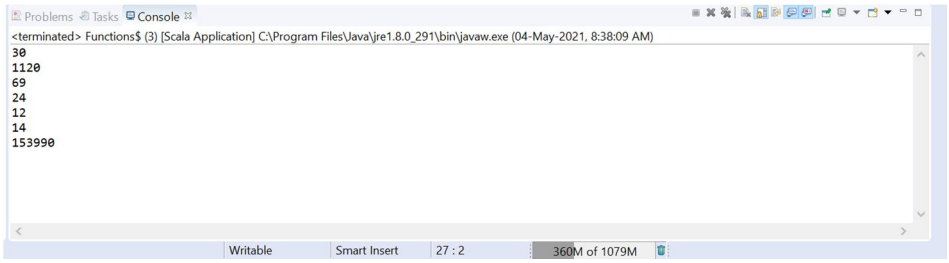


4) Now we'll see 'functions'. Be it any programming language, functions are the soul of programming. Functions can be defined as a general formula or general set of statement(s) that can work in the same way for different set of values. Itis used to save time and space when writing similar lines of code again and again. In this program, we have defined a very simple function of multiplication of 2 numbers. Functions can also be as complex as it could be but dealing with functions is the easiest in Scala, we just wrote 1 line of code to define our function.

We have also worked on recursion; it's not been a day when we talk about functions and we don't talk about recursion. Recursion is a function that calls itself. Generally, the idea of recursion is to reduce the problem and then work on the big problem taking into consideration the smaller problems.

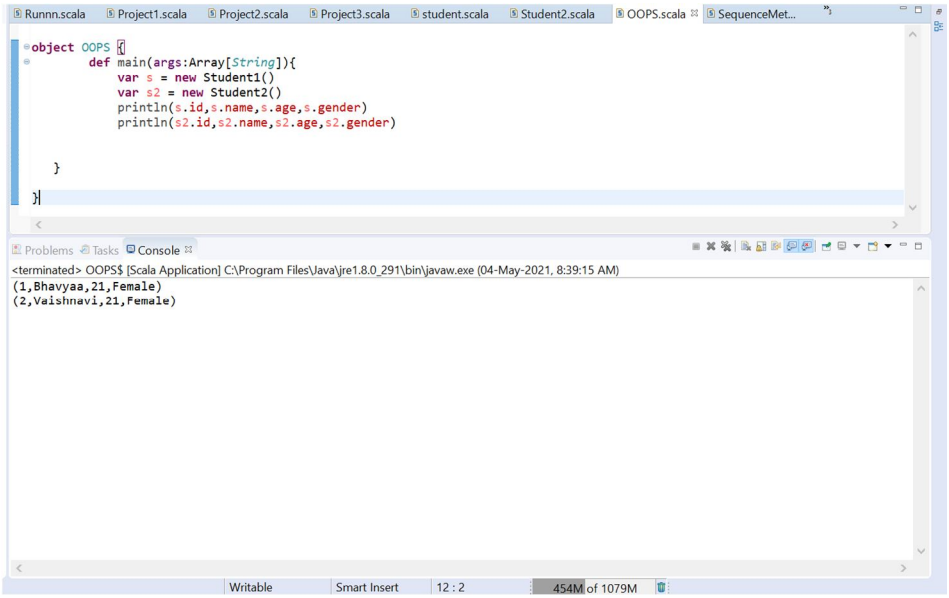


Below is the output of above-mentioned programs:

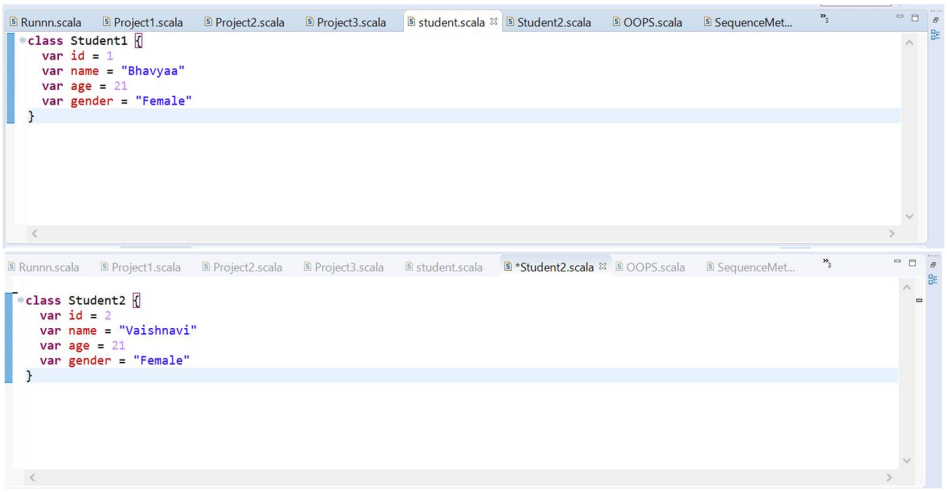


5) Next, we we'll introduce Object- Oriented Programming in Scala:

As shown in the below program, we'll first define new classes with the help of 'new' keyword.

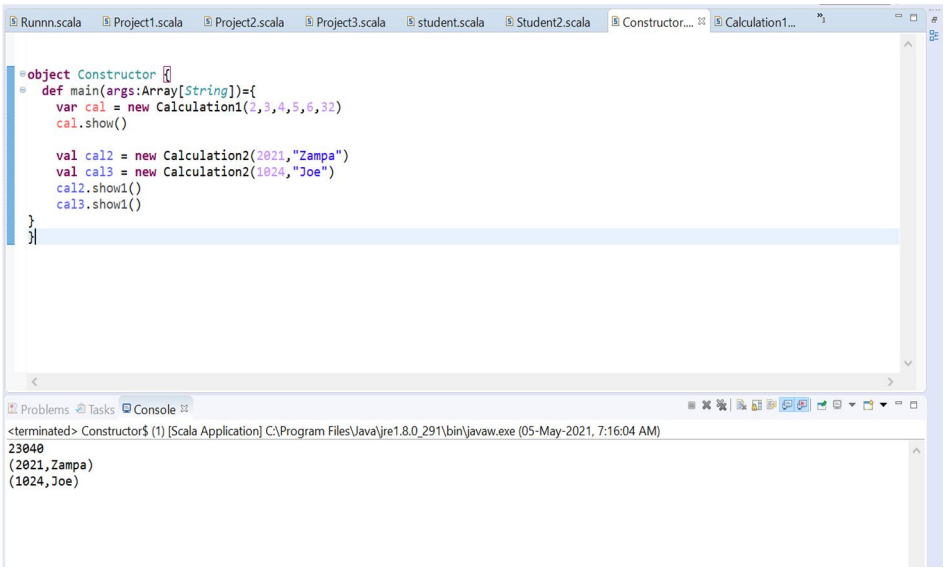


Next, we will define the structure and feature(s) of class(es). In this example we made 2 classes Student1 and Student2, we can have as many of them. This looks so sorted.

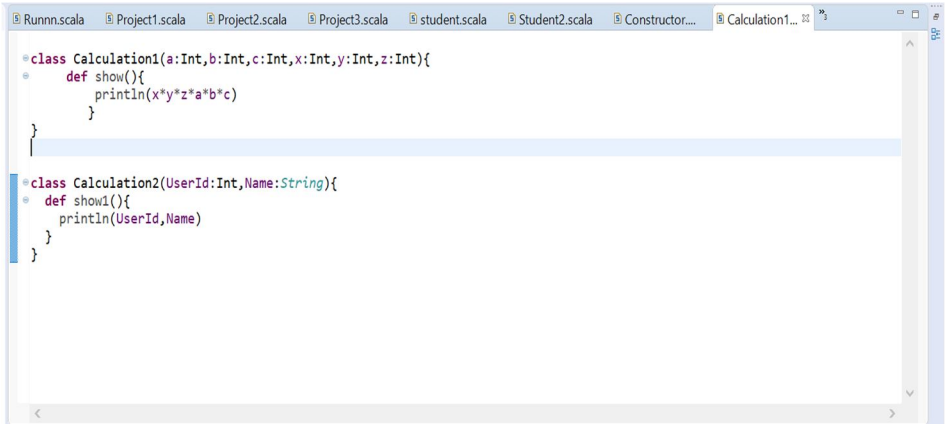


Lastly, we can call and print in the object window. As you can se this looks very neat. One can add features to the individual classes any day without disturbing the whole code.

6) We can now discuss about 'constructors. Constructures is somewhat a mix of functions and object- oriented programming. It is used when one needs to keep things separate, we define the function(s) in one place and print and declare in another.



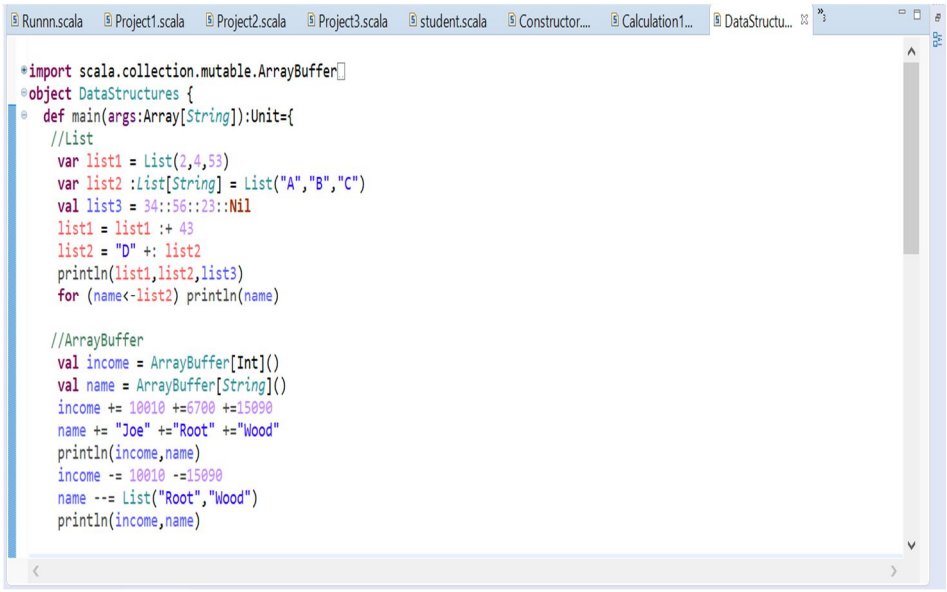
In this example, we used 2 calculations.



7) Data Structures

a) List: This is one of those data structures that we use most commonly. We can define lists of different data types and can perform multiple operations too accordingly. Some of the operations are also depicted in the program.

b) Array Buffer: ArrayBuffer an List are almost same, the difference is ArrayBuffer is a mutable data structure whereas Lists are not mutable in Scala. Also, we can't use different data types in sameArrayBufferwhereas we can do that in Lists, if they are not explicitly restricted to single data type only.

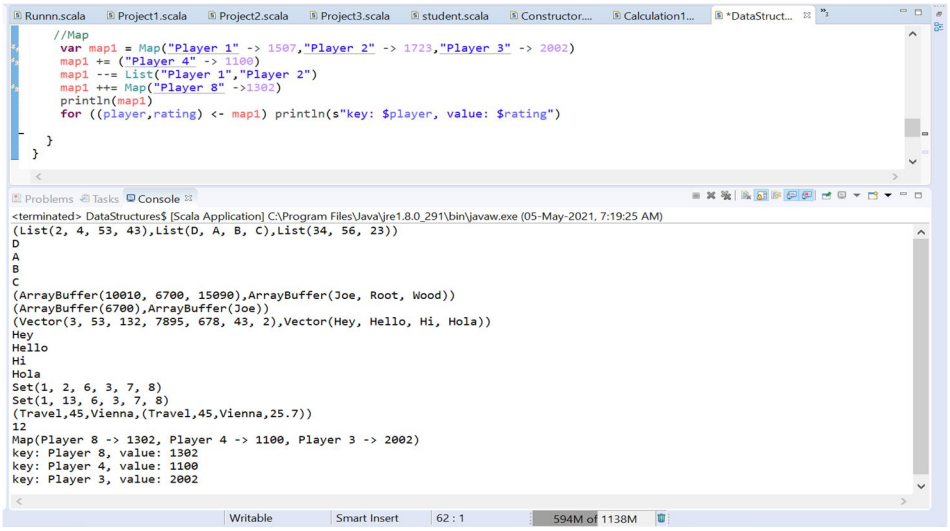


c) Vector: A vector can be defined as one dimensional array to store data. It can store different type of data in the same vector, but it is not mutable. So, we need to operate accordingly on vectors. Some operations are also depicted in the below program.

d) Set: A set is considered to be most useful data structure when we want unique entries from our dataset. Irrespective of how many duplicate entries are passed in a set, it will always return unique values. Sets are again immutable in Scala. Some operations on sets are shown in the following program.

e) Tuple: A tuple is a very useful data structure; it comes handy specially when we need to store data of different entities separately. Fetching data from a tuple is also very easy, it can be seen in the example.

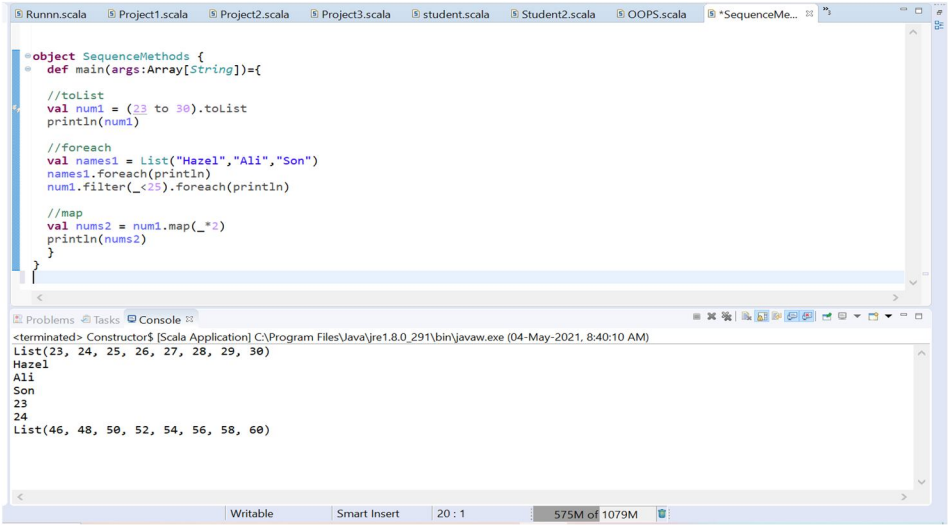
f) *Finally Maps:* Maps are useful when we need to store data in 'key – value' pairs. Maps are also immutable in Scala. We have shown some operations on Map below.



While there are many more data structures in Scala, but these are the very basic and important ones one should know. Also, one should not restrict oneself to the operations we used in above examples, there are many more, and lot of them. As each and every detail cannot be covered in a paper with a restricted scope, one can just get the idea and discover by their own self.

8) Lastly, we will see some sequence generating methods, that can be very useful to use raw as well as in any program to generate a sequence.

To List, foreach and map are some methods we used to generate sequence.



V. CONCLUSION

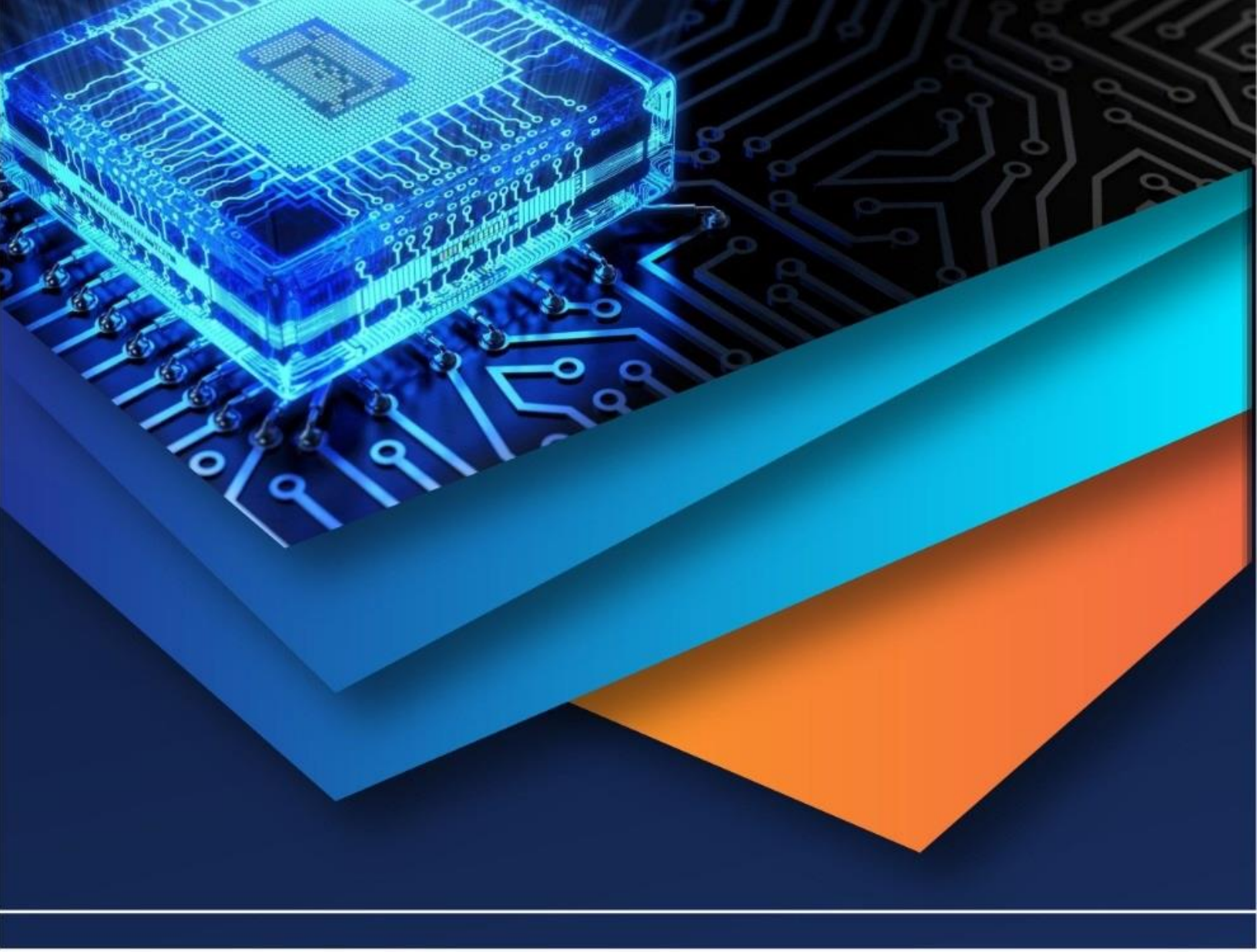
Thus, the huge amount of data generated and collected everyday requires good processing ,flexible and high performing tools to give proper insights. Big Data Analytics is a security enhancing tool of the future. The amount of information we can gather, organize, and apply4 to users in a personalized fashion would take a human, days, weeks, or even months to accomplish. In the capitalistic market such as the United States of America's, competition is key. Time cannot be wasted gathering information and making decisions on incidents that have already taken place. Stopping incidents in their tracks, completing investigative work, and quarantining threatening sources needs to happen immediately and allow for administrators/management to make a on the spot decision. With big data analytics, more educated decisions can be made and focus can remain on business operations moving forward.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

REFERNCES

I have taken help from the following websites to complete this term paper

[1] https://www.sas.com/en_in/insights/big-data/what-is-big-data.html
[2] https://en.wikipedia.org/wiki/Big_data
[3] https://www.edureka.co/blog/what-is-hadoop/
[4] https://www.mongodb.com/why-use-mongodb
[5] https://www.simplilearn.com/what-is-big-data-analytics-article#:~:text=Big%20Data%20analytics%20is%20a,fraudulent%20activities%2C%20among%20other%20things.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY