



# शालेय शिक्षण व क्रीडा विभाग, महाराष्ट्र शासन

शिक्षक / शिक्षण सेवक पदभरती- 2022


स्व- प्रमाणपत्र

Print Date : 09/09/2023

अर्जदाराचे नाव : SHAIKH UBAID AHMAD RAIS AHMAD

परीक्षा क्रमांक : 1210001214

## वैयक्तिक माहिती

अर्जदाराचे नाव	SHAIKH UBAID AHMAD RAIS AHMAD			
लिंग	Male	जन्मतारीख	14/09/2000	
आधार क्रमांक	XXXXXXXX9322	मिळालेले गुण	106	
वैवाहिक स्थिती	Unmarried	ह्यात अपत्यांची संख्या	0	
28/03/2006 नंतर जन्मलेल्या अपत्यांची संख्या	0	जुळे	--	
ईमेल आयडी	ub705856@gmail.com			
अद्ययावत संपर्क क्रमांक	8208307227,9420662297			

## कायमचा पत्ता

राज्य	MAHARASHTRA	जिल्हा	JALGAON
तालुका	JALGAON	गाव	JALGAON
पत्ता	241 MIRZA PAN HOUSE, KATIYA FILE, SHANI PETH, JALGAON. Pin code 425001		

## पत्र व्यवहाराचा पत्ता

राज्य	MAHARASHTRA	जिल्हा	JALGAON
तालुका	JALGAON	गाव	JALGAON
पत्ता	241 MIRZA PAN HOUSE, KATIYA FILE, SHANI PETH, JALGAON. Pin code 425001		



# शालेय शिक्षण व क्रीडा विभाग, महाराष्ट्र शासन

शिक्षक / शिक्षण सेवक पदभरती- 2022

स्व- प्रमाणपत्र

Print Date : 09/09/2023

अर्जदाराचे नाव : SHAIKH UBAID AHMAD RAIS AHMAD

परीक्षा क्रमांक : 1210001214

## जातप्रवर्ग, समांतर आरक्षण व इतर तपशील :

### जातप्रवर्ग तपशील

महाराष्ट्राचे अधिवासी (Domicile)	Yes	अर्जदाराचा जातसंवर्ग	EWS
शारीरिक अपंगत्व	No	भरतीसाठीचा जातसंवर्ग	EWS

### समांतर आरक्षण

महिला	No	माजी सैनिक	No
प्रकल्पग्रस्त	No	भूकंपग्रस्त	No
खेळाडू	No	पदवीधर अंशकालीन कर्मचारी	No
अनाथ	No		

### इतर तपशील

महाराष्ट्र-कर्नाटक सीमा भागातील उमेदवार	No		
आत्महत्याग्रस्त शेतकऱ्याचा मुलगा	No	स्वातंत्र्यसैनिकाचा वारस	No
1994 चे निवडणूक कर्मचारी	No	1991 चे जनगणना कर्मचारी	No

CERTIFIED



# शालेय शिक्षण व क्रीडा विभाग, महाराष्ट्र शासन

शिक्षक / शिक्षण सेवक पदभरती- 2022

स्व- प्रमाणपत्र

Print Date : 09/09/2023

अर्जदाराचे नाव : SHAIKH UBAID AHMAD RAIS AHMAD

परीक्षा क्रमांक : 1210001214

शैक्षणिक अर्हता ( Academic-Qualification)									
अ.क्र	शैक्षणिक अर्हता स्तर	अर्हता	माध्यम	राज्य	विद्यापीठ / मंडळ	निकालाचा दिनांक	गुण	उत्तीर्ण श्रेणी	मुख्य विषय / भाषा
1	SSC (Std. 10th)	S.S.C.	Urdu	Maharashtra	Maharashtra State Board Of Secondary And Higher Secondary Education, Pune	06/06/2016	77.60 %	Distinction	Urdu, Marathi-Hindi, English
2	HSC (Std. 12th)	H.S.C. - Science	English	Maharashtra	Maharashtra State Board Of Secondary And Higher Secondary Education, Pune	30/05/2018	66.00 %	First Division	English, Urdu
Languages Studied in HSC (Arts / Science / Commerce)									
HSC परीक्षा बंगाली भाषा विषय घेऊन उत्तीर्ण आहात काय?				No	HSC परीक्षा कन्नड भाषा विषय घेऊन उत्तीर्ण आहात काय?				No
HSC परीक्षा तेलगू भाषा विषय घेऊन उत्तीर्ण आहात काय?				No	HSC परीक्षा गुजराती भाषा विषय घेऊन उत्तीर्ण आहात काय?				No
HSC परीक्षा सिंधी भाषा विषय घेऊन उत्तीर्ण आहात काय?				No					
* Qualification After 12/02/2023 will not be considered for further process.									



# शालेय शिक्षण व क्रीडा विभाग, महाराष्ट्र शासन

शिक्षक / शिक्षण सेवक पदभरती- 2022

स्व- प्रमाणपत्र

Print Date : 09/09/2023

अर्जदाराचे नाव : SHAIKH UBAID AHMAD RAIS AHMAD

परीक्षा क्रमांक : 1210001214

व्यावसायिक अर्हता (Professional-Qualification)							
अ.क्र	व्यावसायिक अर्हता स्तर	अर्हता	माध्यम	राज्य	विद्यापीठ / मंडळ	निकालाचा दिनांक	गुण
1	Diploma	D.El.Ed.	Urdu	Maharashtra	Maharashtra State Council Of Examination	19/03/2021	82.75 %





# शालेय शिक्षण व क्रीडा विभाग, महाराष्ट्र शासन

शिक्षक / शिक्षण सेवक पदभरती- 2022

स्व- प्रमाणपत्र

Print Date : 09/09/2023

अर्जदाराचे नाव : SHAIKH UBAID AHMAD RAIS AHMAD

परीक्षा क्रमांक : 1210001214

## महाराष्ट्र शिक्षक पात्रता परीक्षा (MAHATET )

TET परीक्षा पात्र आहात काय ? No

## केंद्रीय शिक्षक पात्रता परीक्षा (CTET ) माहिती

CTET परीक्षा पात्र आहात काय ? Yes

पेपर -१		Yes	
अर्जदाराचे नाव (TAIT)	SHAIKH UBAID AHMAD RAIS AHMAD	अर्जदाराचे नाव (CTET)	SHAIKH UBAID AHMAD RAIS AHMAD
शिक्षक पात्रता परीक्षा (CTET ) आसन क्रमांक		172082885410015	
परीक्षेचा महिना आणि वर्ष	December,2021	टक्के (%)	105.00/150.00 (70.00 %)
निकालाची तारीख	09/03/2022	नावामधील विसंगती मंजूर करणारे अधिकारी	
पेपर - २ (Math / Science)		No	
पेपर - २ (Social Science)		No	

CERTIFIED



# शालेय शिक्षण व क्रीडा विभाग, महाराष्ट्र शासन

शिक्षक / शिक्षण सेवक पदभरती- 2022

स्व- प्रमाणपत्र

Print Date : 09/09/2023

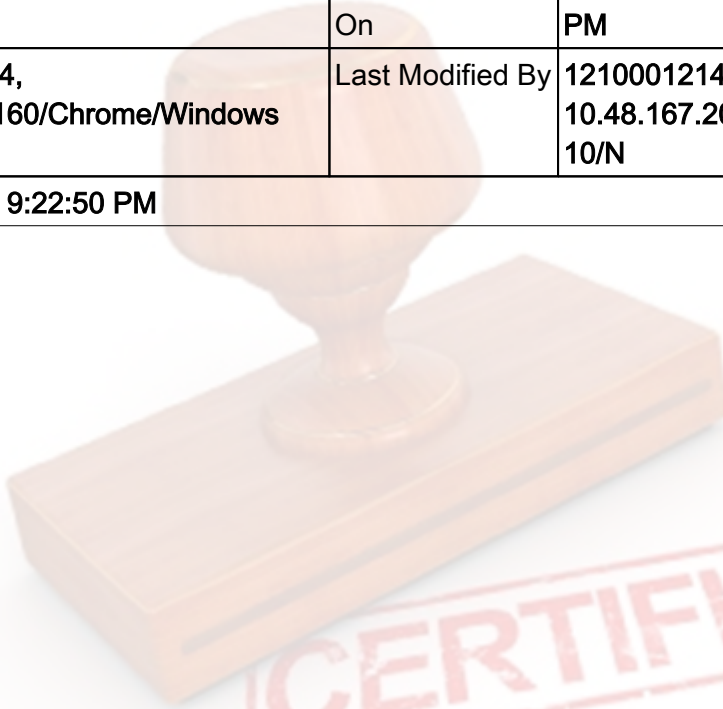
अर्जदाराचे नाव : SHAIKH UBAID AHMAD RAIS AHMAD

परीक्षा क्रमांक : 1210001214

मी अर्जामध्ये भरलेली उपरोक्त माहिती तपासली असून, ती बरोबर असल्याची खात्री केली आहे. भरलेली माहिती व त्यासंबंधीची कागदपत्रे यामध्ये तफावत आढळल्यास मी कायदेशीर कारवाईस पात्र राहीन व माझी निवड कोणत्याही टप्प्यावर रद्द होईल याची मला जाणीव आहे.

(SHAIKH UBAID AHMAD RAIS AHMAD )

Certified On	09/09/2023 09/09/2023 09:22:36 PM	Last Modified On	09/05/2023 05/09/2023 04:18:08 PM
Certified By	1210001214, 152.58.33.160/Chrome/Windows 10/N	Last Modified By	1210001214, 10.48.167.203/Chrome/Windows 10/N
Printed On	09/09/2023 9:22:50 PM		



To inherit a class, you simply incorporate the definition of one class into another by using the extends key word.

```
class subclass-name extends superclass-name { // body of class
}
```

You can only specify one superclass for any subclass that you create. Java does not support the inheritance of multiple superclasses into a single subclass. You can, as stated, create a hierarchy of inheritance in which a subclass becomes a superclass of another subclass. However, no class can be a superclass of itself.

Although a subclass includes all of the members of its superclass, it cannot access those members of the superclass that have been declared as private.

**A Superclass Variable Can Reference a Subclass Object:**

It is important to understand that it is the type of the reference variable—not the type of the object that it refers to—that determines what members can be accessed. When a reference to a subclass object is assigned to a superclass reference variable, you will have access only to those parts of the object defined by the superclass.

```
plainbox    = weightbox;
(superclass) (subclass)
```

```
SUPERCLASS ref = new SUBCLASS(); // HERE ref can only access methods which are available in SUPERCLASS
```

Using super:

Whenever a subclass needs to refer to its immediate superclass, it can do so by use of the keyword super.

super has two general forms. The first calls the superclass' constructor.

The second is used to access a member of the superclass that has been hidden by a member of a subclass.

```
BoxWeight(double w, double h, double d, double m) {
    super(w, h, d); // call superclass constructor
    weight = m;
}
```

Here, BoxWeight( ) calls super( ) with the arguments w, h, and d. This causes the Box constructor to be called,

which initializes width, height, and depth using these values. BoxWeight no longer initializes these values itself.

It only needs to initialize the value unique to it: weight. This leaves Box free to make these values private if desired.

Thus, super( ) always refers to the superclass immediately above the calling class.

This is true even in a multileveled hierarchy.

```
class Box {
    private double width;
    private double height;
    private double depth;
```

```

// construct clone of an object

Box(Box ob) { // pass object to constructor
    width = ob.width;
    height = ob.height;
    depth = ob.depth;
}
}

class BoxWeight extends Box {
    double weight; // weight of box

    // construct clone of an object

    BoxWeight(BoxWeight ob) { // pass object to constructor
        super(ob);
        weight = ob.weight;
    }
}

```

Notice that `super()` is passed an object of type `BoxWeight`—not of type `Box`. This still invokes the constructor `Box(Box ob)`.

NOTE: A superclass variable can be used to reference any object derived from that class.

Thus, we are able to pass a `BoxWeight` object to the `Box` constructor. Of course, `Box` only has knowledge of its own members.

### A Second Use for `super`

The second form of `super` acts somewhat like this, except that it always refers to the superclass of the subclass in which it is used.

`super.member`

Here, `member` can be either a method or an instance variable. This second form of `super` is most applicable to situations in which member names of a subclass hide members by the same name in the superclass.

`super()` always refers to the constructor in the closest superclass. The `super()` in `BoxPrice` calls the constructor in `BoxWeight`. The `super()` in `BoxWeight` calls the constructor in `Box`. In a class hierarchy, if a superclass constructor requires parameters, then all subclasses must pass those parameters “up the line.” This is true whether or not a subclass needs parameters of its own.

If you think about it, it makes sense that constructors complete their execution in order of derivation. Because a superclass has no knowledge of any subclass, any initialization it needs to perform is separate from and possibly prerequisite to any initialization performed by the subclass. Therefore, it must complete its execution first.

NOTE: If `super()` is not used in subclass’ constructor, then the default or parameterless constructor of each superclass will be executed.



Using final with Inheritance:

The keyword final has three uses:

# First, it can be used to create the equivalent of a named constant.

# Using final to Prevent Overriding:

To disallow a method from being overridden, specify final as a modifier at the start of its declaration.

Methods declared as final cannot be overridden.

Methods declared as final can sometimes provide a performance enhancement: The compiler is free to inline calls to them

because it “knows” they will not be overridden by a subclass. When a small final method is called, often the Java

compiler can copy the bytecode for the subroutine directly inline with the compiled code of the calling method, thus

eliminating the costly overhead associated with a method call. Inlining is an option only with final methods

Normally, Java resolves calls to methods dynamically, at run time. This is called late binding. However, since final

methods cannot be overridden, a call to one can be resolved at compile time. This is called early binding.

# Using final to Prevent Inheritance:

Sometimes you will want to prevent a class from being inherited. To do this, precede the class declaration with final.

NOTE: Declaring a class as final implicitly declares all of its methods as final, too.

As you might expect, it is illegal to declare a class as both abstract and final since an abstract class is incomplete

by itself & relies upon its subclasses to provide complete implementations.

# NOTE: Although static methods can be inherited, there is no point in overriding them in child classes because the

method in parent class will run always no matter from which object you call it. That is why static interface methods

cannot be inherited because these methods will run from the parent interface and no matter if we were allowed to

override them, they will always run the method in parent interface.

That is why static interface method must have a body.

NOTE : Polymorphism does not apply to instance variables.