TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

ABU SAYEM
IT-22008

Difference between abstract class and interface

Abstract class	Interface
1. Can have both abstract and non-abstract methods.	1. All methods are abstract by default
2. can have constructors.	2. Cannot have constructors.
3. can have instance variables	3. cannot have instance variables.
4. A class can extend only one abstract class	4. A class can implement multiple interfaces.
5. Uses extends keyword	5. Uses implements Keyword.

TOPIC NAME : DAY :
TIME : DATE : / /

Abe Sayem
It-22008

When to use interface and When to use abstract class (Story & Application)

Imagine you building a system for platform like youtube. Where different kinds of content creator upload different kinds of content. Video, podcast, ~~Blogg~~ blogg etc

Use an abstract class for—
- Shared behaviour across all creator
- A common base type

Use interface for—
- optional abilities
- Behaviour that can be added to any creator

GOOD LUCK™

TOPIC NAME : DAY :
TIME : DATE : / /

Abstract class :

```
abstract class content {
 string name;
 string contenttitle;
public content(string name, string contenttitle){
 this. name = name;
 this contenttitle = contenttitle; }
public void upload content (){
System. out. println(name+" upload; "contenttitle)
}
public abstract void create content ();
}
```

Interface :

```
interface livestreamable {
   void golive (); }
interface monetizable {
   void showAds (); }
interface collaboration {
void. collaborate with (string partnername)
```

"GOOD LUCK"

TOPIC NAME : TIME : DATE : / /

```
class videocreator extends content creator implements
Live streamable, monetizable
public videoCreator(String name, string contenttitle){
Super(name, contenttitle); }
@override
public void creacontent(){
System.out.println(name+" is editing"); }
public void golive(){
System.out.println(name+"is live now!"); }
public void shorads(){
System.out.println(name+"is sharing ads"); }
public void acceptSponsorship(){
System.out.println(name+" accepted a sponsorship
class Blogger extends contentcreator implement
Collaborator {
public Blogger(String name, string Contenttitl
{
super(name, contenttitle); }
public void createContent(){
System.out.println(name+" is writing"); }
```

TOPIC NAME : DAY : TIME : DATE : / /

```
public void collaborateWith(String partnerName)
{
System.out.println(name + " is collaborating with
    " + partnerName + " on a blog.");
}
}
```