14.Android event handling

android වලදී button එකක් press කිරම event එකක් නම් වේ.press කල විට සිදු විය යුතු දෙයට අපි කියන්නේ handler කියලයි.එනම් එම සිදු වූ event එක handle කල යුත්තේ කෙසේද කියල අපි ඒ අදාළ handler method එකේ ලිවිය යුතුයි.Events කියන දේවල් නිතරම වන්නේ නෑ.එම නිසා අපි ඒවා සිදු වෙනකන් අහන් සිටිය යුතුයි.එම නිසා එම ක්‍රියාවලියට කියනවා event listning කියල.අපි දන්නවා,අපි ශබ්ද අහන්නේ කනෙන්ය,රූප දක්කිනේ ඇස් දෙකෙන්ය.ශීතල/රස්නය දැනගන්නේ සමෙන්ය.මෙලස ඒ ඒ events සඳහා ඒවා අහන් ඉදල තේරුම් ගතහැකි විවද listners තියෙනවා කියල අපිට තේරෙනවා.android වලද මෙසේම ඒ ඒ events අහන් ඉන්න event listners බොහොමයක් පවතී.

android වල ඉහත කියන ලද event listners, Interface ටිකක් ලෙස ගොඩ නගා තියෙනවා.එනම් අපි මෙම interface එක implement කරන class එකක් සෑදා එයින් object එකක් සෑදිය යුතුයි.එනම් එම object එකෙන් තමා event listen කරන්නේ.**So android වල events listen කිරීම සඳහා object එකක් අපි සෑදිය යුතුයි.එම object එක හදන class එක, සුදුසු listner interface එකක් implement කර තිබිය යුතුයි.**

තවද, හරියටම කියනවනම් මේ interfaces ඇත්තේ View class එක තුලයි.සාමාන්‍යයෙන් අපි දන්නවා interfaces, class වලින් පිට වෙනමයි code කරන්නේ.යම් class එකකට interface implement කිරීමයි කල හැක්කේ.නමුත් මේ event listner interface ටික View class එක තුල තියෙන නිසා ඒවා internal/nested interfaces බව අපිට තේරුම් ගන්න පුළුවන්.තවත් දැනගත යුතු දෙයක් වන්නේ මෙම interfaces තුල ඇත්තේ එක method එකක් පමණි.එලෙසම අපි දන්නවා ඕනෑම interface එකක් තුල ඇත්තේ abstract methods පමණි.So මෙම event listner interface තුල තිබෙන එකම එක method එක abstract method එකක් වේ.මේ abstract method එක වන්නේ එම event listner එකට අදාළ Event Handler method එකයි.**So event listne කිරීමට සාදන object එකේ class එක සඳහා listner interface එකක් implement කිරීමෙන් ලබන්නේ එක method එකයි.එය event handler abstract method එක වේ.**

එනම්, events listen කරන් ඉන්න කියල හදල තියෙන interface එක තුල තියෙන්නේ,එම event එක සිදු වුනහම සිදු විය යුතු method එකයි.මේ method එක abstract method එකක් වේ.එනම් එම method එක ඇතුලේ code කසිවක් නෑ,method එකේ නම පමණයි.එය එසේ විය යුතුයි ඉතින්.මොකද අපිට කැමති විදියටයි, සිදු විය යුතු දෙය code කරන්නේ.තවත් විශේෂ දෙයක් වන්නේ මෙම interface එක තුල තිබෙන එකම method එක abstract method එකක් පමණක් නොව, එය callback method එකක් ලෙසටද භාවිතා කරන්න කියලයි android අපිට කියන්නේ.

programming වලදී callback method එකක් යනු,

සිතන්න යම් විශේෂ method එකක්/process එකක් තියෙනවා කියල.අපේ පහසුවට අපි එම method එක A ලෙස නම් කර ගමු.තවද මෙම method එක ඉවර වූ සැනින් සිදු විය යුතු තව method එකක් තියෙනවා සිතන්න.එනම් A නම් method එක අවසන් වන තෙක් පටන් ගත නොහැකි method එකක් ඇතියි සිතන්න.අපේ පහසුවට අපි එය B ලෙස නම් කර ගමු.එවිට programming වල ක්‍රමයක් තියෙනවා,අපිට code කරලා තියන්න පුලුවන් A කියන method එක කොයි වෙලේ හෝ අවසන් වුනහම B method එක සිදු වෙන්න කියල.සාමන්‍යයෙන් අපිට හැගෙන්නේ, B method එක සිදු වෙන්න කියල කියල code කල යුත්තේ A method එක අවසානයේදීයි කියලයි.නමුත් ඊට වඩා හොඳ ක්‍රමයක් ඇත.එය නම් අපිට පුළුවන් B method එකේ address එක (පළමු address එක) A method එකට pass කරන්න argument එකක් ලෙස.එනම් A method එකේ parameter එකක් තිබිය යුතුයි function එකක address එකක් අල්ලා ගත හැකි.so function එකක address එකක් ලෙස තේරුම් අරන් රදවා තබා ගත හැකි variable type එක වන්නේ function type එකම වේ.පහත psudo code වලින් ලියා ඇති උදාහරණය බැලූ විට මෙය හොඳින් පැහැදිලි වේ.

```
function B(){
        print("world");
}
function A(function callBck){
        print ("hello");
}
function main(){
        A(B);
}
```

ඉහතදී call back method එක වන්නේ B() යන්නයි.මොකද එය යම් තවත් විශේෂ method එකක්/process එකක් සිදු වුන් පසු එම විශේෂ process එක හෝ method එක මගින්ම B() method එක සිදු වෙන්න කියලා කියන නිසයි.එම නිසයි ඒවාට call back (මම ඉවර වුනාට පස්සේ ඔයාට කතා කරන්නම්) methods කියන්නේ.නමුත් java වල ඉහත පරිදි call back methods කියල දෙයක් හරියටම නෑ.මොකද object oriented language වල තියෙන්නේ නිකන්ම methods නෙමේ,object වලට අදාළ methods වේ.so java වල call back method වලින් සිදු වන දෙය කරන්නේ object හරහාය.එනම් ඉහතදී අපි pass කලේ යම් method එකක address එකයි.නුමුත් java වල කරන්නේ object එකක reference එක pass කිරීමයි.නමුත් සරලව සිදු වන ක්‍රියාවලිය එකම වගෙයි.යමක් වූ සැනින් තව දෙයක් කිරීමයි.

so event handling සඳහා මෙම සංකල්පය යොදා ගත හැක.එනම් event එකක් වූ සැනින් යමක් කිරීම සඳහා. **Android framework එක අපි හදපු object එක හරහා events අහන් ඉන්නවා.event එකක් සිදු වූ සැනින් event handler method එකට call කරනවා.So අපි event handler method එක තියෙන object එකේ reference එක framework එකට pass කල යුතුයි.එවිටයි එය listen කරන් ඉදලා event එක සිදු වූ ගමන් එම object එකේ handler method එක fire කල හැක්කේ.විශේෂත්වය නම් android වල මෙම event handler method එක තියෙන්නෙත් events listen කරන object එක තුල වීමයි.**

So, Event listner nested interface වල තියෙන එකම method එක abstract method එකක් හා callback method එකක් වේ.එය event handler method එක වේ.අපි listener interface එක implement කරන class එක තුල මෙම abstract handler method එක, code type කර implement කල යුතුයි.

so මේ සියල්ල සරලව ගත් කල,events listen කිරීම සඳහා object එකක් සැදිය යුතුයි.ඒ සඳහා අපිට අවශ්‍ය listner interface එක implement කර class එකක් සැදිය යුතුයි.මෙම interface එකේ තිබෙන එකම method එක වන්නේ event එක සිදු විට fire වන method එක නිසා,එය මෙම class එක තුල implement කල යුතුයි,අපිට අවශ්‍ය දේවල් සිදු වන ලෙස.ඉන් පසු අපි මෙම object එක register කල යුතුයි අවශ්‍ය ui element එකත් සමග.ඒ සඳහා setOnClickListner() වැනි methods use කල හැක.තවද මෙම method එකට අපි pass කල යුතුයි argument එකක් ලෙස event handler method එක තියෙන object එකේ reference එක.වාසනවකට එය listner object එකම වේ.මොකද android වල listening හා handling දෙකම එක object එකකින් කරන ලෙසයි code කල යුත්තේ.

දැන් අපි විවිද listner interface වර්ග හා ඒවා තුල ඇති handler method එක මොනවාදයි බලමු.

Event Listner	Event Handler
OnClickListner	onClick()
OnLongClickListener- button,text,image වැනි ඕනෑම widget එකක් තත්පරයකට වඩා press කරන් ඉන්න events listen කිරීම සඳහා.	onLongClick()
OnFocusChangeListener	onFocusChange()
OnFocusChangeListener	onKey()
OnTouchListener	onTouch()
OnMenuItemClickListener	onMenuClick()
OnCreateContextMenuItemListener	onCreateContextMenuItem()

Events Handle කිරීම කල හැකි ක්‍රම

දැන් අපි දන්නවා යම් view එකක් මත සිදු කරන event එකක් listen කිරීම සඳහා **සහ** එම event එක handle කරීම සඳහා අපි **එක** class එකක් සාදා එහි object **එකක්** සැදිය යුතුයි කියල.ඉන් පසුව ui element එකේ අදාළ event එක listen කිරීම සඳහා ඉහත සදා ගත් object එක ui element එකට set කල යුතුයි. event එකක් සිදු වූ විට එම ui element එකේ object reference එකක් throw කරනවා handler method එක හට.එම නිසා handler method එකේ parameter එකක් තිබිය යුතුයි ui elements වල object referece එකක් ගබඩා කර ගත හැකි.සියලු ui elements View class එකේ sub class නිසා, parameter එකේ type එක View ලෙස තැබීම සුදුසුයි.

1.Nested class එකක් මගින්.එනම් Listening සහ handling කරන class එකක් MainActivity class එක තුලම සාදා.

```
public class MainActivity extends Activity{

        @override
        public void onCreate(Bundle savedInstanceState){
                super.onCreate(savedInstanceState);

                setContentView(R.layout.activity_main);

                Button b1=(Button)findViewById(R.id.button1);

                b1.setOnClickListner(new ListenAndHandle());

        }

        private class ListenAndHandle implements OnClickListener{

                public void onClick(View v){

                        // button1 යන button එක click කල විට සිදු විය යුතු දේවල්.
                }
        }
}
```

ඉහතදී අපි ප්‍රධාන class එක තුල එනම් ප්‍රධාන activity එකට අදාළ class එක තුලම තව class එකක් සාදා තිබේ.අපිට මෙලස class එකක් තුල තව class එකක් සැදුව හැකියි.ඒවාට අපි nested classes කියල කියනවා.යම් class එකකින් එක object එකක් පමණයි නම් අවශ්‍ය අපි බොහෝ වෙලාවට ඒවා inner classes ලෙසයි code කරන්නේ.මොකද මෙහිදී අපිට අවශ්‍ය එක object එකයි events listen කිරීමට හා එය handle කිරීමට.එම නිසා listener interface එක implement කරන එක class එකක් සාදා, එහි object එකක් සාදා, එම object එකේ reference එක, button එකට listner object එක set කරන විට pass කරා නම් හරි.

ඉහත ක්‍රමයේදී object එක සාදන එක හා reference එක pass කරන එක එක විටම කර ඇත.මොකද setOnClickListener() යන method එකට යවා ඇති argument එක වන්නේ new ListenAndHandle() යන්නයි. ListenAndHandle() යනු inner class එකේ constructor method එක වේ.new මගින් කරන්නේ constructor method එක ලවා අලුත් method එකක් සාදවා,එම සෑදුන object එකේ address එක return කිරීමයි.object එකේ address එක යනු එහි reference එකයි.So මෙම ක්‍රමයේදීත් වන්නේ **handler method එක තියෙන object එකේ reference එකක්, button එකට listner object එක set කරන විට pass වීමයි.**

2.OnClickListener interface එක implement කරන anonymous class එකක් මගින්

anonymous classes,inner classes,local classes යන සියලු වර්ග nested classes වර්ග වේ.එම නිසා මෙම ක්‍රම වලදී බොහෝ විට අලුත් class එක (listen කිරීම හා handling කිරම සඳහා වන class එක) define කරන්නේ Activity class එක තුලම වේ.

```
public class MainActivity extends Activity{

        @override
        public void onCreate(Bundle savedInstanceState){
                super.onCreate(savedInstanceState);

                setContentView(R.layout.activity_main);

                Button b1=(Button)findViewById(R.id.button1);

                b1.setOnClickListner(ListenAndHandle);

        }

        private OnClickListener ListenAndHandle = new OnClickListener{

                public void onClick(View v){

                        // button1 යන button එක click කල විට සිදු විය යුතු දේවල්.
                }
        };
}
```

3.Anonymous Inner Class

අපි දන්නවා events listen කිරීමට object එකක් තිබිය යුතු බව.සහ view element එකට එම event listner එක set කරන විට එය තුලට argument එකක් ලෙස එම event listener object එකේ reference එකත් pass කල යුතුයි.ඉතින් අපිට anonymous class සංකල්පය භාවිතා කරමින්,argument එක යවන තැන class එක define කරමින් object එක සාදා එම object එකේ reference එක pass කල හැක.

```
public class MainActivity extends Activity{

        @override
        public void onCreate(Bundle savedInstanceState){
                super.onCreate(savedInstanceState);

                setContentView(R.layout.activity_main);

                Button b1=(Button)findViewById(R.id.button1);

                b1.setOnClickListner(new OnClickListener{

                        public void onClick(View v){

                                // button1 යන button එක click කල විට සිදු විය යුතු දේවල්.
                        }
                });
        }
}
```

4.Activity Class එක තුල listener interface එක implement කර.

```
public class MainActivity extends Activity implements OnClickListener{

        @override
        public void onCreate(Bundle savedInstanceState){
                super.onCreate(savedInstanceState);

                setContentView(R.layout.activity_main);

                Button b1=(Button)findViewById(R.id.button1);

                b1.setOnClickListner(this)


                public void onClick(View v){

                        // button1 යන button එක click කල විට සිදු විය යුතු දේවල්.
                }
        }
}
```