



## COMPSCI 280 S1 2011 Enterprise Software Development

Developing Windows Applications with Visual Studio 2010  
Debugging Applications (Extra Reference Only)



## Agenda & Reading

- ▶ **Agenda:**
  - ▶ Three Types of Errors
  - ▶ Debugging Setting
  - ▶ Code Stepping
  - ▶ Breakpoints
  - ▶ Break Mode
  - ▶ Viewing Data in the Debugger
    - ▶ DataTips
    - ▶ Variable Windows
    - ▶ Visualizers
    - ▶ Immediate Window
  - ▶ Tracepoints
- ▶ **Recommended Reading:**
  - ▶ Debugging in Visual Studio
    - ▶ [http://msdn2.microsoft.com/en-us/library/sc65sadd\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/sc65sadd(VS.80).aspx)
- ▶ **Hands-On Lab:**
  - ▶ Lecture15Lab

2

COMPSCI280

Handout15



## Three Types of Errors

- ▶ **Three types of errors can occur in a VB program:**
  - ▶ **Syntax errors**
    - ▶ Syntax Error (compile error) is a programming mistake that violates the rules of VB.
    - ▶ VB points out syntax error while you type program statements
    - ▶ VB won't let you run a program until you fix all syntax errors.
  - ▶ **Run-time errors**
    - ▶ A run-time error is a mistake that causes a program to stop unexpectedly during execution
  - ▶ **Logic errors**
    - ▶ A logic error is a human error – a programming mistake that makes the program code produce the wrong results.
    - ▶ You can use debugging efforts to track down the logic errors

3

COMPSCI280

Handout15



## Debugging

- ▶ The Visual Studio debugger is a powerful tool that allows you to observe the run-time behavior of your program and locate logic errors.
- ▶ With the debugger, you can break, or suspend, execution of your program to
  - ▶ examine your code,
  - ▶ evaluate and edit variables in your program
- ▶ The debugger works with several files in order to cross-reference memory to the source files used to build the assembly
  - ▶ A program database (PDB) file holds debugging and project state information that allows incremental linking of a debug configuration of your program.
  - ▶ To debug your code, the debugger needs PDB files containing symbols for your application

4

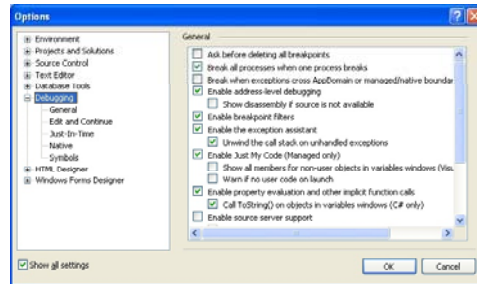
COMPSCI280

Handout15



## Debugger Settings

- ▶ You can specify various settings for debugger behavior,
  - ▶ including how variables are displayed,
  - ▶ whether certain warnings are presented,
  - ▶ how breakpoints are set, and
  - ▶ how breaking affects running programs.
- ▶ To set debugger options
  - ▶ Choose Tools -> Options.
  - ▶ Expand Debugging



5

COMPSCI280

Handout15



## Debugger Roadmap

- ▶ Code Stepping
  - ▶ Step Into
  - ▶ Step Over
  - ▶ Step Out
- ▶ Breakpoints
  - ▶ Insert Breakpoint
  - ▶ Break at Function
- ▶ Break Mode
  - ▶ Run to Cursor
  - ▶ Call Stack
- ◆ Viewing Data in the Debugger
  - DataTips
  - Variable Windows
    - ◆ Locals
    - ◆ Autos
    - ◆ Watch
    - ◆ Quick Watch
  - Visualizers
  - Immediate Window
- ◆ Tracepoints

6

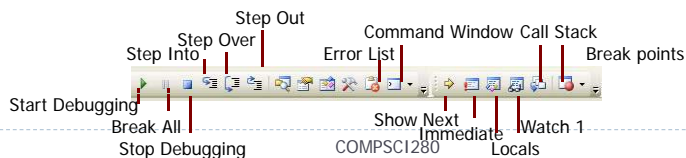
COMPSCI280

Handout15



## Code Stepping

- ▶ One of the most common debugging procedures is stepping:
  - ▶ executing code one line at a time.
- ▶ Choose Debug -> Step Into/Over/Out
  - ▶ Step Into/Over
    - ▶ Instruct the debugger to execute the next line of code
    - ▶ Note:
      - Step Into executes only the call itself for function call, then halts at the first line of code inside the function.
      - Step Over executes the entire function, then halts at the first line outside the function.
  - ▶ Step Out resumes execution of your code until the function returns, then breaks at the return point in the calling function.



7

COMPSCI280

Handout15



## Breakpoints

- ▶ Breakpoints tell the debugger that an application should break, pause execution, at a certain point.
- ▶ When a break occurs, your program and the debugger are said to be in break mode
- ▶ To set a simple breakpoint
  - ▶ Click on a line of executable code in the source window
  - ▶ Choose Debug->Toggle Breakpoint, or
  - ▶ Right-click, choose Breakpoint->Insert Breakpoint
- ▶ To start/stop debugging
  - ▶ Choose Debug->Start/Stop Debugging
- ▶ To stop debugging and restart
  - ▶ Choose Debug->Restart.
    - ▶ Note: Restart stops the current debugging session and restarts the startup project.

8

COMPSCI280

Handout15



## Breakpoints (con't)

- ▶ **To insert a function breakpoint**
  - ▶ In a source window, click on the name of a function.
  - ▶ Choose Debug->New Breakpoint->Break at Function.
- ▶ **To set a breakpoint on a function call**
  - ▶ In the Call Stack window, right-click the function call
  - ▶ Choose Insert Breakpoint
- ▶ **To delete a breakpoint**
  - ▶ In the Breakpoints window, right-click on a breakpoint, and choose Delete from the shortcut
- ▶ **To disable/enable a breakpoint**
  - ▶ Right-click on a line containing the breakpoint
  - ▶ Choose Disable/Enable Breakpoint from the shortcut menu



## Break Mode

- ▶ **Breaks execution:**
  - ▶ When execution reaches a breakpoint
  - ▶ When an exception occurs
  - ▶ Break execution manually at any time
    - ▶ Choose Debug->Break All
  - ▶ Break at the location where the cursor is set.
    - ▶ In a source window, right-click on a line of executable code, and
    - ▶ Right-click, choose Run to Cursor
  - ▶ Break at a specified function
    - ▶ Choose Debug->Windows->Call Stack
    - ▶ In the Call Stack window, right-click the function name and choose Run To Cursor from the shortcut menu.



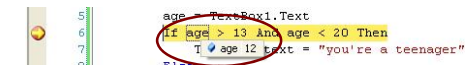
## Viewing Data in the Debugger

- ▶ The Visual Studio debugger provides a variety of tools for inspecting and modifying the state of your program.
- ▶ Most of these tools function only in break mode.
  - ▶ DataTips
  - ▶ Variable Windows
    - ▶ Locals
    - ▶ Autos
    - ▶ Watch
  - ▶ QuickWatch dialog box
  - ▶ Visualizers
  - ▶ Immediate Window



## DataTip

- ▶ **Conditions**
  - ▶ DataTips work only in break mode, and
  - ▶ Only with variables that are within the current scope of execution.
- ▶ **To display a DataTip**
  - ▶ Choose Debug->StepInTo
    - ▶ Enter break mode
    - ▶ In a source window, place the mouse pointer over any variable in the current scope.
- ▶ **Expanding Information**
  - ▶ With enhanced DataTips, you can expand an array, structure, or object to view its members.
  - ▶ To expand a variable to see its elements
    - ▶ Hover the mouse cursor over the + sign that precedes the variable name.
    - ▶ Use the arrow keys on your keyboard to move up and down



```

Dim i() As Integer = N
For i = 0 To a.Length-1
  Con (0) 1 i.WriteLine
  Next (1) 3
  Sub (2) 5
  (3) 7
  (4) 9
  (5) 11
  (6) 13
  (7) 15
  (8) 17
  (9) 19
  (10) 21
  (11) 23
  (12) 25
  (13) 27
  (14) 29
  (15) 31
  (16) 33
  (17) 35
  (18) 37
  (19) 39
  (20) 41
  (21) 43
  (22) 45
  (23) 47
  (24) 49
  (25) 51
  (26) 53
  (27) 55
  (28) 57
  (29) 59
  (30) 61
  (31) 63
  (32) 65
  (33) 67
  (34) 69
  (35) 71
  (36) 73
  (37) 75
  (38) 77
  (39) 79
  (40) 81
  (41) 83
  (42) 85
  (43) 87
  (44) 89
  (45) 91
  (46) 93
  (47) 95
  (48) 97
  (49) 99
  (50) 101
  (51) 103
  (52) 105
  (53) 107
  (54) 109
  (55) 111
  (56) 113
  (57) 115
  (58) 117
  (59) 119
  (60) 121
  (61) 123
  (62) 125
  (63) 127
  (64) 129
  (65) 131
  (66) 133
  (67) 135
  (68) 137
  (69) 139
  (70) 141
  (71) 143
  (72) 145
  (73) 147
  (74) 149
  (75) 151
  (76) 153
  (77) 155
  (78) 157
  (79) 159
  (80) 161
  (81) 163
  (82) 165
  (83) 167
  (84) 169
  (85) 171
  (86) 173
  (87) 175
  (88) 177
  (89) 179
  (90) 181
  (91) 183
  (92) 185
  (93) 187
  (94) 189
  (95) 191
  (96) 193
  (97) 195
  (98) 197
  (99) 199
  (100) 201
  (101) 203
  (102) 205
  (103) 207
  (104) 209
  (105) 211
  (106) 213
  (107) 215
  (108) 217
  (109) 219
  (110) 221
  (111) 223
  (112) 225
  (113) 227
  (114) 229
  (115) 231
  (116) 233
  (117) 235
  (118) 237
  (119) 239
  (120) 241
  (121) 243
  (122) 245
  (123) 247
  (124) 249
  (125) 251
  (126) 253
  (127) 255
  (128) 257
  (129) 259
  (130) 261
  (131) 263
  (132) 265
  (133) 267
  (134) 269
  (135) 271
  (136) 273
  (137) 275
  (138) 277
  (139) 279
  (140) 281
  (141) 283
  (142) 285
  (143) 287
  (144) 289
  (145) 291
  (146) 293
  (147) 295
  (148) 297
  (149) 299
  (150) 301
  (151) 303
  (152) 305
  (153) 307
  (154) 309
  (155) 311
  (156) 313
  (157) 315
  (158) 317
  (159) 319
  (160) 321
  (161) 323
  (162) 325
  (163) 327
  (164) 329
  (165) 331
  (166) 333
  (167) 335
  (168) 337
  (169) 339
  (170) 341
  (171) 343
  (172) 345
  (173) 347
  (174) 349
  (175) 351
  (176) 353
  (177) 355
  (178) 357
  (179) 359
  (180) 361
  (181) 363
  (182) 365
  (183) 367
  (184) 369
  (185) 371
  (186) 373
  (187) 375
  (188) 377
  (189) 379
  (190) 381
  (191) 383
  (192) 385
  (193) 387
  (194) 389
  (195) 391
  (196) 393
  (197) 395
  (198) 397
  (199) 399
  (200) 401
  (201) 403
  (202) 405
  (203) 407
  (204) 409
  (205) 411
  (206) 413
  (207) 415
  (208) 417
  (209) 419
  (210) 421
  (211) 423
  (212) 425
  (213) 427
  (214) 429
  (215) 431
  (216) 433
  (217) 435
  (218) 437
  (219) 439
  (220) 441
  (221) 443
  (222) 445
  (223) 447
  (224) 449
  (225) 451
  (226) 453
  (227) 455
  (228) 457
  (229) 459
  (230) 461
  (231) 463
  (232) 465
  (233) 467
  (234) 469
  (235) 471
  (236) 473
  (237) 475
  (238) 477
  (239) 479
  (240) 481
  (241) 483
  (242) 485
  (243) 487
  (244) 489
  (245) 491
  (246) 493
  (247) 495
  (248) 497
  (249) 499
  (250) 501
  (251) 503
  (252) 505
  (253) 507
  (254) 509
  (255) 511
  (256) 513
  (257) 515
  (258) 517
  (259) 519
  (260) 521
  (261) 523
  (262) 525
  (263) 527
  (264) 529
  (265) 531
  (266) 533
  (267) 535
  (268) 537
  (269) 539
  (270) 541
  (271) 543
  (272) 545
  (273) 547
  (274) 549
  (275) 551
  (276) 553
  (277) 555
  (278) 557
  (279) 559
  (280) 561
  (281) 563
  (282) 565
  (283) 567
  (284) 569
  (285) 571
  (286) 573
  (287) 575
  (288) 577
  (289) 579
  (290) 581
  (291) 583
  (292) 585
  (293) 587
  (294) 589
  (295) 591
  (296) 593
  (297) 595
  (298) 597
  (299) 599
  (300) 601
  (301) 603
  (302) 605
  (303) 607
  (304) 609
  (305) 611
  (306) 613
  (307) 615
  (308) 617
  (309) 619
  (310) 621
  (311) 623
  (312) 625
  (313) 627
  (314) 629
  (315) 631
  (316) 633
  (317) 635
  (318) 637
  (319) 639
  (320) 641
  (321) 643
  (322) 645
  (323) 647
  (324) 649
  (325) 651
  (326) 653
  (327) 655
  (328) 657
  (329) 659
  (330) 661
  (331) 663
  (332) 665
  (333) 667
  (334) 669
  (335) 671
  (336) 673
  (337) 675
  (338) 677
  (339) 679
  (340) 681
  (341) 683
  (342) 685
  (343) 687
  (344) 689
  (345) 691
  (346) 693
  (347) 695
  (348) 697
  (349) 699
  (350) 701
  (351) 703
  (352) 705
  (353) 707
  (354) 709
  (355) 711
  (356) 713
  (357) 715
  (358) 717
  (359) 719
  (360) 721
  (361) 723
  (362) 725
  (363) 727
  (364) 729
  (365) 731
  (366) 733
  (367) 735
  (368) 737
  (369) 739
  (370) 741
  (371) 743
  (372) 745
  (373) 747
  (374) 749
  (375) 751
  (376) 753
  (377) 755
  (378) 757
  (379) 759
  (380) 761
  (381) 763
  (382) 765
  (383) 767
  (384) 769
  (385) 771
  (386) 773
  (387) 775
  (388) 777
  (389) 779
  (390) 781
  (391) 783
  (392) 785
  (393) 787
  (394) 789
  (395) 791
  (396) 793
  (397) 795
  (398) 797
  (399) 799
  (400) 801
  (401) 803
  (402) 805
  (403) 807
  (404) 809
  (405) 811
  (406) 813
  (407) 815
  (408) 817
  (409) 819
  (410) 821
  (411) 823
  (412) 825
  (413) 827
  (414) 829
  (415) 831
  (416) 833
  (417) 835
  (418) 837
  (419) 839
  (420) 841
  (421) 843
  (422) 845
  (423) 847
  (424) 849
  (425) 851
  (426) 853
  (427) 855
  (428) 857
  (429) 859
  (430) 861
  (431) 863
  (432) 865
  (433) 867
  (434) 869
  (435) 871
  (436) 873
  (437) 875
  (438) 877
  (439) 879
  (440) 881
  (441) 883
  (442) 885
  (443) 887
  (444) 889
  (445) 891
  (446) 893
  (447) 895
  (448) 897
  (449) 899
  (450) 901
  (451) 903
  (452) 905
  (453) 907
  (454) 909
  (455) 911
  (456) 913
  (457) 915
  (458) 917
  (459) 919
  (460) 921
  (461) 923
  (462) 925
  (463) 927
  (464) 929
  (465) 931
  (466) 933
  (467) 935
  (468) 937
  (469) 939
  (470) 941
  (471) 943
  (472) 945
  (473) 947
  (474) 949
  (475) 951
  (476) 953
  (477) 955
  (478) 957
  (479) 959
  (480) 961
  (481) 963
  (482) 965
  (483) 967
  (484) 969
  (485) 971
  (486) 973
  (487) 975
  (488) 977
  (489) 979
  (490) 981
  (491) 983
  (492) 985
  (493) 987
  (494) 989
  (495) 991
  (496) 993
  (497) 995
  (498) 997
  (499) 999
  (500) 1001
  (501) 1003
  (502) 1005
  (503) 1007
  (504) 1009
  (505) 1011
  (506) 1013
  (507) 1015
  (508) 1017
  (509) 1019
  (510) 1021
  (511) 1023
  (512) 1025
  (513) 1027
  (514) 1029
  (515) 1031
  (516) 1033
  (517) 1035
  (518) 1037
  (519) 1039
  (520) 1041
  (521) 1043
  (522) 1045
  (523) 1047
  (524) 1049
  (525) 1051
  (526) 1053
  (527) 1055
  (528) 1057
  (529) 1059
  (530) 1061
  (531) 1063
  (532) 1065
  (533) 1067
  (534) 1069
  (535) 1071
  (536) 1073
  (537) 1075
  (538) 1077
  (539) 1079
  (540) 1081
  (541) 1083
  (542) 1085
  (543) 1087
  (544) 1089
  (545) 1091
  (546) 1093
  (547) 1095
  (548) 1097
  (549) 1099
  (550) 1101
  (551) 1103
  (552) 1105
  (553) 1107
  (554) 1109
  (555) 1111
  (556) 1113
  (557) 1115
  (558) 1117
  (559) 1119
  (560) 1121
  (561) 1123
  (562) 1125
  (563) 1127
  (564) 1129
  (565) 1131
  (566) 1133
  (567) 1135
  (568) 1137
  (569) 1139
  (570) 1141
  (571) 1143
  (572) 1145
  (573) 1147
  (574) 1149
  (575) 1151
  (576) 1153
  (577) 1155
  (578) 1157
  (579) 1159
  (580) 1161
  (581) 1163
  (582) 1165
  (583) 1167
  (584) 1169
  (585) 1171
  (586) 1173
  (587) 1175
  (588) 1177
  (589) 1179
  (590) 1181
  (591) 1183
  (592) 1185
  (593) 1187
  (594) 1189
  (595) 1191
  (596) 1193
  (597) 1195
  (598) 1197
  (599) 1199
  (600) 1201
  (601) 1203
  (602) 1205
  (603) 1207
  (604) 1209
  (605) 1211
  (606) 1213
  (607) 1215
  (608) 1217
  (609) 1219
  (610) 1221
  (611) 1223
  (612) 1225
  (613) 1227
  (614) 1229
  (615) 1231
  (616) 1233
  (617) 1235
  (618) 1237
  (619) 1239
  (620) 1241
  (621) 1243
  (622) 1245
  (623) 1247
  (624) 1249
  (625) 1251
  (626) 1253
  (627) 1255
  (628) 1257
  (629) 1259
  (630) 1261
  (631) 1263
  (632) 1265
  (633) 1267
  (634) 1269
  (635) 1271
  (636) 1273
  (637) 1275
  (638) 1277
  (639) 1279
  (640) 1281
  (641) 1283
  (642) 1285
  (643) 1287
  (644) 1289
  (645) 1291
  (646) 1293
  (647) 1295
  (648) 1297
  (649) 1299
  (650) 1301
  (651) 1303
  (652) 1305
  (653) 1307
  (654) 1309
  (655) 1311
  (656) 1313
  (657) 1315
  (658) 1317
  (659) 1319
  (660) 1321
  (661) 1323
  (662) 1325
  (663) 1327
  (664) 1329
  (665) 1331
  (666) 1333
  (667) 1335
  (668) 1337
  (669) 1339
  (670) 1341
  (671) 1343
  (672) 1345
  (673) 1347
  (674) 1349
  (675) 1351
  (676) 1353
  (677) 1355
  (678) 1357
  (679) 1359
  (680) 1361
  (681) 1363
  (682) 1365
  (683) 1367
  (684) 1369
  (685) 1371
  (686) 1373
  (687) 1375
  (688) 1377
  (689) 1379
  (690) 1381
  (691) 1383
  (692) 1385
  (693) 1387
  (694) 1389
  (695) 1391
  (696) 1393
  (697) 1395
  (698) 1397
  (699) 1399
  (700) 1401
  (701) 1403
  (702) 1405
  (703) 1407
  (704) 1409
  (705) 1411
  (706) 1413
  (707) 1415
  (708) 1417
  (709) 1419
  (710) 1421
  (711) 1423
  (712) 1425
  (713) 1427
  (714) 1429
  (715) 1431
  (716) 1433
  (717) 1435
  (718) 1437
  (719) 1439
  (720) 1441
  (721) 1443
  (722) 1445
  (723) 1447
  (724) 1449
  (725) 1451
  (726) 1453
  (727) 1455
  (728) 1457
  (729) 1459
  (730) 1461
  (731) 1463
  (732) 1465
  (733) 1467
  (734) 1469
  (735) 1471
  (736) 1473
  (737) 1475
  (738) 1477
  (739) 1479
  (740) 1481
  (741) 1483
  (742) 1485
  (743) 1487
  (744) 1489
  (745) 1491
  (746) 1493
  (747) 1495
  (748) 1497
  (749) 1499
  (750) 1501
  (751) 1503
  (752) 1505
  (753) 1507
  (754) 1509
  (755) 1511
  (756) 1513
  (757) 1515
  (758) 1517
  (759) 1519
  (760) 1521
  (761) 1523
  (762) 1525
  (763) 1527
  (764) 1529
  (765) 1531
  (766) 1533
  (767) 1535
  (768) 1537
  (769) 1539
  (770) 1541
  (771) 1543
  (772) 1545
  (773) 1547
  (774) 1549
  (775) 1551
  (776) 1553
  (777) 1555
  (778) 1557
  (779) 1559
  (780) 1561
  (781) 1563
  (782) 1565
  (783) 1567
  (784) 1569
  (785) 1571
  (786) 1573
  (787) 1575
  (788) 1577
  (789) 1579
  (790) 1581
  (791) 1583
  (792) 1585
  (793) 1587
  (794) 1589
  (795) 1591
  (796) 1593
  (797) 1595
  (798) 1597
  (799) 1599
  (800) 1601
  (801) 1603
  (802) 1605
  (803) 1607
  (804) 1609
  (805) 1611
  (806) 1613
  (807) 1615
  (808) 1617
  (809) 1619
  (810) 1621
  (811) 1623
  (812) 1625
  (813) 1627
  (814) 1629
  (815) 1631
  (816) 1633
  (817) 1635
  (818) 1637
  (819) 1639
  (820) 1641
  (821) 1643
  (822) 1645
  (823) 1647
  (824) 1649
  (825) 1651
  (826) 1653
  (827) 1655
  (828) 1657
  (829) 1659
  (830) 1661
  (831) 1663
  (832) 1665
  (833) 1667
  (834) 1669
  (835) 1671
  (836) 1673
  (837) 1675
  (838) 1677
  (839) 1679
  (840) 1681
  (841) 1683
  (842) 1685
  (843) 1687
  (844) 1689
  (845) 1691
  (846) 1693
  (847) 1695
  (848) 1697
  (849) 1699
  (850) 1701
  (851) 1703
  (852) 1705
  (853) 1707
  (854) 1709
  (855) 1711
  (856) 1713
  (857) 1715
  (858) 1717
  (859) 1719
  (860) 1721
  (861) 1723
  (862) 1725
  (863) 1727
  (864) 1729
  (865) 1731
  (866) 1733
  (867) 1735
  (868) 1737
  (869) 1739
  (870) 1741
  (871) 1743
  (872) 1745
  (873) 1747
  (874) 1749
  (875) 1751
  (876) 1753
  (877) 1755
  (878) 1757
  (879) 1759
  (880) 1761
  (881) 1763
  (882) 1765
  (883) 1767
  (884) 1769
  (885) 1771
  (886) 1773
  (887) 1775
  (888) 1777
  (889) 1779
  (890) 1781
  (891) 1783
  (892) 1785
  (893) 1787
  (894) 1789
  (895) 1791
  (896) 1793
  (897) 1795
  (898) 1797
  (899) 1799
  (900) 1801
  (901) 1803
  (902) 1805
  (903) 1807
  (904) 1809
  (905) 1811
  (906) 1813
  (907) 1815
  (908) 1817
  (909) 1819
  (910) 1821
  (911) 1823
  (912) 1825
  (913) 1827
  (914) 1829
  (915) 1831
  (916) 1833
  (917) 1835
  (918) 1837
  (919) 1839
  (920) 1841
  (921) 1843
  (922) 1845
  (923) 1847
  (924) 1849
  (925) 1851
  (926) 1853
  (927) 1855
  (
```



## Variable Windows

- ▶ The debugger provides a number of variable windows for displaying, evaluating, and editing variables and expressions.
- ▶ Each variable window includes a grid with three columns: Name, Value, and Type.
  - ▶ The Name column contains variable names or expressions.
  - ▶ The Value and Type column display the value and data type of the variable or expression.
- ▶ Variable windows
  - ▶ Locals
  - ▶ Autos
  - ▶ Watch



## Locals Windows

- ▶ **Locals Window**
  - ▶ It displays variables **local** to the current context or scope.
  - ▶ Usually, this means the procedure or function you are currently executing.
  - ▶ Note: the debugger populates this window automatically.
- ▶ **To open the Locals window**
  - ▶ Enter break mode
  - ▶ Choose Debug->Windows->Locals
- ▶ **To expand a variable to see its elements**
  - ▶ Hover the mouse cursor over the + sign that precedes the variable name.

Name	Value	Type
Me	{Debugging.Form1}	Debugging.Form1
a	{Length=5}	Integer()
age	12	Integer
e	{System.Windows.Forms.Mou	System.EventArgs
i	0	Integer
p	{Length=2}	System.Drawing.Point()
sender	{System.Windows.Forms.Butt	Object



## Autos Windows

- ▶ **Autos Window**
  - ▶ It displays variables used in the **current** line of code and the preceding line of code.
  - ▶ Again, the debugger populates this window automatically.
- ▶ **To open the Autos window**
  - ▶ Enter break mode
  - ▶ Choose Debug->Windows->Autos

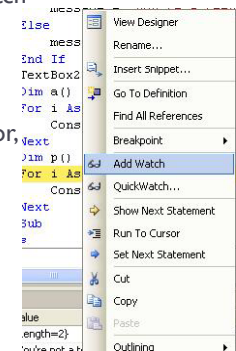
Name	Value	Type
a	{Length=2}	Integer()
p	{Length=2}	System.Drawing.Point()
X (0)	{X = 10 Y = 20}	System.Drawing.Point
En	{X = 0 Y = 0}	System.Drawing.Point
Isf	False	Boolean
X	10	Integer
Y	20	Integer
X (1)	{X = 20 Y = 40}	System.Drawing.Point



## Watch Windows

- ▶ **Watch window**
  - ▶ It is where you can add variables whose value you want to watch.
  - ▶ You can add variables and any valid expression
  - ▶ You can open up to four watch windows, numbered Watch 1, Watch 2, Watch 3, and Watch 4.
- ▶ **To open the watch window**
  - ▶ Enter break mode
  - ▶ Select the variable you want to add from the Code editor: right click on it, and click the Add Watch command
  - ▶ Choose Debug->Windows->Watch 1

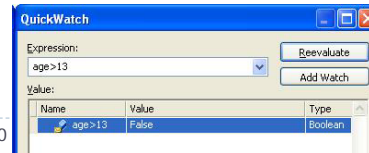
Name	Value	Type
p	{Length=2}	System.Drawing.Point()
message	"You're not a teenager"	String





## QuickWatch dialog box

- ▶ The QuickWatch dialog box is similar in concept to the Watch window, but QuickWatch can display only one variable or expression at a time.
- ▶ Useful when you want to take a quick look at a variable or expression without bringing up the Watch window.
- ▶ QuickWatch is a modal dialog box, you have to close it before you can continue debugging
- ▶ To open the QuickWatch window
  - ▶ Enter break mode
  - ▶ Select the variable you want to add from the Code editor, right click on it, and click the Quick Watch command to open the QuickWatch Dialog box
  - ▶ To add a QuickWatch expression to the Watch window
    - ▶ Click the Add Watch button
    - ▶ Enter the expression and press Enter.



17

COMPSCI280



## Visualizers

- ▶ Visualizers are a new component of the Visual Studio debugger user interface.
- ▶ A visualizer creates a dialog box or other interface to displays a variable or object
- ▶ The debugger come with four standard visualizers:
  - ▶ text,
  - ▶ HTML,
  - ▶ XML, and
  - ▶ dataset visualizer
- ▶ Visualizers are represented by a magnifying glass icon
- ▶ To open the Visualizer
  - ▶ Enter break mode
  - ▶ Open one of the variable windows ( Locals/Auto/Watch/QuickWatch)
  - ▶ Click on the magnifying glass to select a visualizer appropriate to the data type of the corresponding object.



18

COMPSCI280

Handout15



## Immediate Window

- ▶ The Immediate window is used to debug and evaluate expressions, execute statements, print variable values, and so forth.
- ▶ It allows you to enter expressions to be evaluated or executed by the development language during debugging.
- ▶ To display the Immediate window
  - ▶ Enter break mode
  - ▶ Choose Debug->Windows->Immediate.
  - ▶ Type the following in the Immediate window in a break mode
- ▶ To modify a variable using the Immediate window
  - ▶ Enter break mode
  - ▶ Choose Debug->Windows->Immediate
  - ▶ Type "age=15" and press Enter
  - ▶ Watch the changes in the Watch I window

```
?age<Enter>
?message<Enter>
?age+3<Enter>
```



19

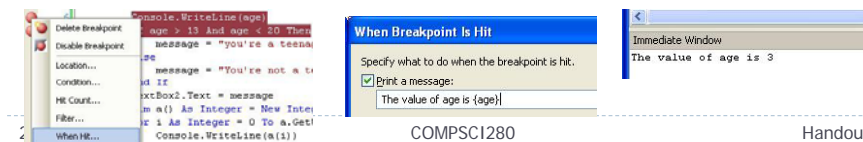
COMPSCI280

Handout15



## Tracepoints

- ▶ A tracepoint is a breakpoint with a custom action associated with it.
  - ▶ When the tracepoint is hit, it causes the debugger to perform the specified action instead of (or in addition to) breaking program execution.
  - ▶ The message is written to the Immediate window.
- ▶ Create a tracepoint in two different ways:
  - ▶ Convert an existing breakpoint into a tracepoint by adding a tracepoint action.
  - ▶ Create a tracepoint from scratch with the New Tracepoint command.
- ▶ To set the action
  - ▶ Right-click the tracepoint, choose When Hit
  - ▶ Choose Print a message



20

COMPSCI280

Handout15