# Test Plans for Atmosphere Program

* VectorTest.cpp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Description (and why) | Actual Test Data | Expected Output | Pass/Fail |
| 1 | Default constructor sets size and capacity to 0. | N/A – Default constructor called. | Size = 0  Capacity = 0 | Pass |
| 2 | Parameterized constructor sets initial capacity. | Vector<int> v(5) | Size = 0  Capacity = 5 | Pass |
| 3 | PushBack() adds elements correctly and updates size. | Vector<int> v(5)  v.PushBack(1) | V[0] = 1  Size = 1  Capacity = 5 | Pass |
| 4 | PushBack() increases capacity when needed. | Vector<int> v(4)  v.PushBack(1)  v.PushBack(2)  v.PushBack(3) | V[0] = 1  V[1] = 2  V[2] = 3  Size = 3  Capacity = 8 | Pass |
| 5 | Copy constructor copies all elements, size, and capacity. | Vector<int> v1(5)  v1.PushBack(1)  v1.PushBack(2)  Vector<int> v2(v1) | V1[0] = 1  V1[1] = 2  Size = 2  Capacity = 5  V2[0] = 1  V2[1] = 2  Size = 2  Capacity = 5 | Pass |
| 6 | Assignment operator copies all elements, size, and capacity. | Vector<int> v1(5)  v1.PushBack(1)  v1.PushBack(2)  Vector<int> v2 = v1 | V1[0] = 1  V1[1] = 2  Size = 2  Capacity = 5  V2[0] = 1  V2[1] = 2  Size = 2  Capacity = 5 | Pass |
| 7 | Subscript ([]) operator returns reference to element. | Vector<int> v1(5)  v1.PushBack(1) | Memory address and values of V[0] are printed correctly | Pass |
| 8 | GetSize() returns the number of inserted elements. | Vector<int> v1(5)  v1.PushBack(1)  v1.PushBack(2) | Size = 2 | Pass |
| 9 | GetCapacity() returns correct capacity after expansion. | Vector<int> v1(5)  v1.PushBack(1)  v1.PushBack(2) | Capacity = 5 | Pass |
| 10 | Copy() in Assignment Operator and Copy Constructor creates a deep copy of the Vector object. | Vector<int> v1(5)  v1.PushBack(1)  v1.PushBack(2)  Vector<int> v2 = v1  Vector<int> v3(v1)  V1[0] = 99 | Memory addresses of the same elements (v1[0], v2[0], v3[0]) are different. The value of v1 at index 0 is different to v2 and v3. | Pass |
| 11 | Vector object is copied correctly when passed by value. | Vector<int> v1(5)  v1.PushBack(1)  V\_copy[0] = 2 (in function) | V\_copy[0] = 2  V1[0] = 1 | Pass |
| 12 | Vector object is correctly modified when passed by non-const reference. | Vector<int> v1(5)  v1.PushBack(1)  V\_ ref[0] = 2 (in function) | V\_ ref[0] = 2  V1[0] = 1 | Pass |
| 13 | Vector object data can be accessed but not modified when passed by const reference. | Vector<int> v1(5)  v1.PushBack(1)  V\_ const\_ref[0] = 2 (in function) | Compiler prints an error when setter in PassByConstRef() is left uncommented and called. | Pass |
| 14 | Vector is correctly returned from a function by value. | Vector object in ReturnObjectCopy()  Vector<int> v(5)  v.PushBack(1) | V[0] In Function: 1  V[0] From Return Function: 1 | Pass |
| 15 | Accessing element at an index greater than size using subscript ([]) operator throws an assertion error. | Vector<int> v(5)  v.PushBack(1)  v[3] = 2 | Compiler prints an error when setter in “v[3] = 2” is left uncommented and called. | Pass |
| 16 | PushBack() handles insertion correctly when capacity is 0 (ie. Default constructor is called for new vector object). | Vector<int> v  v.PushBack(1) | V[0] = 1  Size: 1  Capacity: 3 | Pass |
| 17 | Read, store and print integer data from a file. | 45  2213  235  877  1215  321445  32167  4332 | 45  2213  235  877  1215  321445  32167  4332 | Pass |
| 18 | Read, store and print Date data from a file. | 01/01/2020  15/03/2021  31/12/2019  29/02/2024  04/07/2022  10/10/2010  25/12/2025  30/06/2023  11/11/2011  05/05/2005 | 01/01/2020  15/03/2021  31/12/2019  29/02/2024  04/07/2022  10/10/2010  25/12/2025  30/06/2023  11/11/2011  05/05/2005 | Pass |
| 19 | Read, store and print Unit data from a file. | ICT100 IntroProgramming 3  ICT159 FundamentalsCS 3  ICT283 DataStructures 3  ICT374 Algorithms 3  ICT289 SoftwareEng 3  ICT207 WebDev 3  ICT302 CyberSecurity 3  ICT115 Databases 3  ICT203 OOP 3  ICT210 OSFundamentals 3 | ICT100 IntroProgramming 3  ICT159 FundamentalsCS 3  ICT283 DataStructures 3  ICT374 Algorithms 3  ICT289 SoftwareEng 3  ICT207 WebDev 3  ICT302 CyberSecurity 3  ICT115 Databases 3  ICT203 OOP 3  ICT210 OSFundamentals 3 | Pass |
| 20 | Constructing a vector using capacity < 0 will not work. | Vector<int> v(-1) | Compiler prints an error when improper constructor is left uncommented and called. | Pass |

* DateTest.cpp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Description (and why) | Actual Test Data | Expected Output | Pass/Fail |
| 1 | Default constructor sets marks to 0. | N/A – Default constructor called. | Day: 1  Month: 1  Year: 1 | Pass |
| 2 | Parameterized constructor correctly sets day, month and year. | Day: 13  Month: 3  Year: 2005 | Day: 13  Month: 3  Year: 2005 | Pass |
| 3 | SetDay() correctly updates the day. | Old Day: 13  New Day: 14 | Day Before Setting: 13  Day After Setting: 14 | Pass |
| 4 | GetDay() correctly returns the updated day. | Day: 13 | Day: 13 | Pass |
| 5 | SetMonth() correctly updates the month. | Old Month: 3  New Month: 4 | Month Before Setting: 3  Month After Setting: 4 | Pass |
| 6 | GetMonth() correctly returns the updated month. | Month: 3 | Month: 3 | Pass |
| 7 | SetYear() correctly updates the year. | Old Year: 2005  New Year: 2025 | Year Before Setting: 2005  Year After Setting: 2025 | Pass |
| 8 | GetYear() correctly returns the updated year. | Year: 2005 | Year: 2005 | Pass |
| 9 | SetDay() sets an invalid day due to lack of validation checking. | Old Day: 13  New Day: 33 | Day: 33 | Pass |
| 10 | SetMonth() sets an invalid month due to lack of validation checking. | Old Month: 3  New Month: 15 | Month: 15 | Pass |
| 11 | SetYear() sets an invalid year due to lack of validation checking. | Old Year: 2005  New Year: -2005 | Year: -2005 | Pass |
| 12 | Date object is copied correctly when passed by value. | D1 Date Object  Day: 13  Month: 3  Year: 2005  D\_copy Date Object (pass by value)  Day: 13  Month: 3  Year: 2005  D\_ copy modified Date Object  Day: 15  Month: 3  Year: 2005 | D1 and D\_copy (after modification) print different days  Day of d\_copy: 15  Day of d1: 13 | Pass |
| 13 | Date object is correctly modified when passed by non-const reference. | D1 Date Object  Day: 13  Month: 3  Year: 2005  D\_ref Date Object (pass by reference)  Day: 13  Month: 3  Year: 2005  D\_ref modified Date Object  Day: 15  Month: 3  Year: 2005 | D1 and D\_copy (after modification) print same days  Day of d\_copy: 15  Day of d1: 15 | Pass |
| 14 | Date object data can be safely accessed without modification when passed by const reference. | D1 Date Object  Day: 13  Month: 3  Year: 2005  D\_const\_ref Date Object  Day: 13  Month: 3  Year: 2005  D\_const\_ref attempts to change the day to 14 in the function | Compiler prints an error when setter in PassByConstRef() is left uncommented and called. | Pass |
| 15 | Date object is correctly copied and returned from a function via return-by-value. | Date Object in ReturnObjectCopy()  Day: 13  Month: 3  Year: 2005 | Day In Function: 13  Day From Return Function: 13 | Pass |

* MyTimeTest.cpp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Description (and why) | Actual Test Data | Expected Output | Pass/Fail |
| 1 | Default constructor sets hour and minute to 0. | N/A – Default constructor called. | Time: 0:0 | Pass |
| 2 | Parameterized constructor stores hour and minute. | Hour: 4  Minute: 35 | Time: 4:35 | Pass |
| 3 | SetHour() updates hour value. | Old Hour: 4  New Hour: 12 | Hour Before Setting: 4  Hour After Setting: 12 | Pass |
| 4 | SetMinute() updates minute value. | Old Minute: 35  New Minute: 50 | Minute Before Setting: 35  Minute After Setting: 50 | Pass |
| 5 | GetHour() correctly returns the hour value. | Hour: 4 | Hour: 4 | Pass |
| 6 | GetMinute() correctly returns the minute value. | Minute: 35 | Minute: 35 | Pass |
| 7 | Can store arbitrary values outside 24h format. | Hour: 123  Minute: 90 | Time: 123:90 | Pass |
| 8 | Time object is copied correctly when passed by value. | T1 MyTime Object  Hour: 4  Minute: 35  T\_copy MyTime Object (pass by value)  Hour: 4  Minute: 35  T\_ copy modified MyTime Object  Hour: 5  Minute: 45 | Time in T\_copy: 5:45  Time in T1: 4:35 | Pass |
| 9 | Time object is correctly modified when passed by non-const reference. | T1 MyTime Object  Hour: 4  Minute: 35  T\_ref MyTime Object (pass by reference)  Hour: 4  Minute: 35  T\_ ref modified MyTime Object  Hour: 5  Minute: 45 | Time in T\_ ref: 5:45  Time in T1: 5:45 | Pass |
| 10 | Time object data can be safely accessed without modification when passed by const reference. | T1 MyTime Object  Hour: 4  Minute: 35  T\_const\_ref MyTime Object  Hour: 4  Minute: 35  T\_const\_ref attempts to change the hour to 14 in the function | Compiler prints an error when setter in PassByConstRef() is left uncommented and called. | Pass |
| 11 | Time object is correctly copied and returned from a function via return-by-value. | MyTime Object in ReturnObjectCopy()  Hour: 4  Minute: 35 | Time in Function: 4:35  Time From Return Function: 4:35 | Pass |