

```
1: // Object.h - Header file for java.lang.Object.  -*- c++ -*-
2:
3: /* Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation
4:
5:    This file is part of libgcj.
6:
7: This software is copyrighted work licensed under the terms of the
8: Libgcj License. Please consult the file "LIBGCCJ_LICENSE" for
9: details.  */
10:
11: #ifndef __JAVA_LANG_OBJECT_H__
12: #define __JAVA_LANG_OBJECT_H__
13:
14: #pragma interface
15:
16: #include <gcj/javaprims.h>
17:
18: // This class is mainly here as a kludge to get G++ to allocate two
19: // extra entries in each vtable.
20: struct _JvObjectPrefix
21: {
22: protected:
23:     // New ABI Compatibility Dummy, #1 and 2.
24:     virtual void nacd_1 (void) {}; // This slot really contains the Class
pointer.
25:     // For IA64, the GC descriptor goes into the second word of the nacd1 descr.
26:     #ifndef __ia64__
27:         virtual void nacd_2 (void) {}; // Actually the GC bitmap marking
descriptor.
28:     #endif
29: };
30:
31: class java::lang::Object : public _JvObjectPrefix
32: {
33: protected:
34:     virtual void finalize (void);
35: public:
36:     // Order must match order in Object.java.
37:     jclass getClass (void);
38:     virtual jint hashCode (void);
39:     void notify (void);
40:     void notifyAll (void);
41:     void wait (jlong timeout, jint nanos);
42:     virtual jboolean equals (jobject obj);
43:     Object (void);
44:     virtual jstring toString (void);
45:     void wait (void);
46:     void wait (jlong timeout);
47:
48:     friend void _Jv_MonitorEnter (jobject obj);
49:     friend void _Jv_MonitorExit (jobject obj);
50:     friend void _Jv_InitializeSyncMutex (void);
51:     friend void _Jv_FinalizeObject (jobject obj);
52:     friend bool _Jv_ObjectCheckMonitor (jobject obj);
53:
54: #ifdef JV_MARKOBJ_DECL
55:     friend JV_MARKOBJ_DECL;
56: #endif
57: #ifdef JV_MARKARRAY_DECL
58:     friend JV_MARKARRAY_DECL;
59: #endif
60:
```

```
61:     static java::lang::Class class$;
62:
63: protected:
64:     virtual jobject clone (void);
65:
66: private:
67:     // This does not actually refer to a Java object. Instead it is a
68:     // placeholder for a piece of internal data (the synchronization
69:     // information).
70:     #ifndef JV_HASH_SYNCHRONIZATION
71:         jobject sync_info;
72:     #endif
73:
74:     // Initialize the sync_info field. Not called with
JV_HASH_SYNCHRONIZATION.
75:     void sync_init (void);
76: };
77:
78: #endif /* __JAVA_LANG_OBJECT_H__ */
```